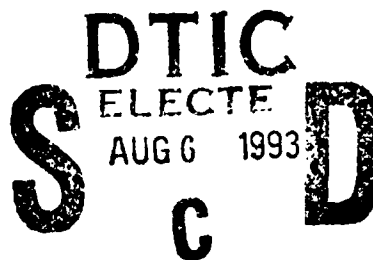


AFIT/DS/ENG/93-05

AD-A267 623



APPLICATION OF SEQUENCE COMPARISON METHODS  
TO MULTISENSOR DATA FUSION  
AND TARGET RECOGNITION

DISSERTATION

Presented to the Faculty of the School of Engineering  
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the  
Requirements for the Degree of  
Doctor of Philosophy

Edmund W. Libby, M.S.  
LTC, USA

18 June, 1993

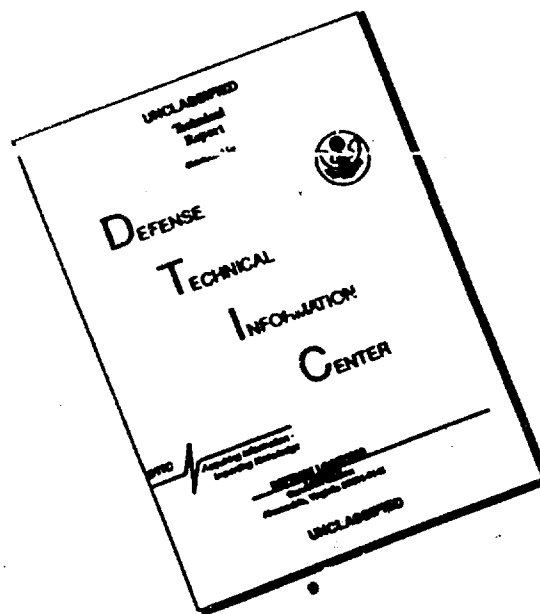
Approved for public release; distribution unlimited

93 8 05 450

93-18097



# DISCLAIMER NOTICE



**THIS DOCUMENT IS BEST  
QUALITY AVAILABLE. THE COPY  
FURNISHED TO DTIC CONTAINED  
A SIGNIFICANT NUMBER OF  
PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.**

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No 0704-0188

This report was prepared for the Department of Defense under contract number DAH04-93-1-0001. It is being made available to the public in accordance with the provisions of the Department of Defense Policy Directive 5, dated 22 July 1993, and the Office of Management and Budget Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE July 1993		3. REPORT TYPE AND DATES COVERED Doctoral Dissertation	
4. TITLE AND SUBTITLE  APPLICATION OF SEQUENCE COMPARISON METHODS TO MULTISENSOR DATA FUSION AND TARGET RECOGNITION				5. FUNDING NUMBERS	
6. AUTHOR(S)  Edmund W. Libby					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Air Force Institute of Technology Wright Patterson AFB, OH 45433-6583				8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT/DS/ENG/93-05	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Wright Laboratory Target Recognition Technology Branch WL/AARA (Mr. Rick Mitchell) Wright-Patterson AFB, OH 45433				10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION / AVAILABILITY STATEMENT  Approved for Public Release. Distribution Unlimited				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  This research addresses methods for exploiting the joint likelihood of observed kinematic and nonkinematic (sensor signature) physical events to improve dynamic object and target recognition. A principal direction is the application of dynamic programming sequence comparison techniques to condition matching of object signatures to known models according to observed kinematics. A second direction is the application of kinematic/aspect-angle Kalman filter trackers to condition kinematic tracking according to observed signatures. These conditioning processes dramatically reduce ambiguity in object recognition, and can be used together or separately to allow computation of a posteriori probabilities of object class membership using Bayesian methods. Proposals are supported by simulated target tracking and high range resolution radar target recognition. Original contributions of this effort include: (1) new approaches for and theoretical understanding of syntactic methods in multisensor fusion and dynamic object recognition; and (2) extension of estimation and tracking techniques to allow object recognition and establish performance bounds for recognition.					
14. SUBJECT TERMS Target Recognition      Kalman Filtering Dynamic Programming      Maximum Likelihood Estimation Radar Tracking      Multisensors				15. NUMBER OF PAGES	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT  UL		

### *Acknowledgements*

Many people have gone out of their way to help me complete this effort, and I am indebted to all. A complete list of people who brought me to this point would be about two pages long -- it would start with my dear mother, to whom I hope someday to be able to explain the meaning of the word "stochastic."

Most importantly, this effort would never have begun, let alone have been completed, without the support of my Committee Chairman, Professor Peter S. Maybeck. His inspiring lectures, writings, and previous research provided a solid foundation for this research, and a boundless resource of ideas to apply against the problems I addressed.

Many other members of the AFIT faculty and staff had key roles in this effort: notably the members of my committee, MAJ Steven K. Rogers and Dr. Bruce Suter, as well as Dr. Matthew Kabrisky, who introduced me to speech processing.

The models used to conduct this research are the result of many years of painstaking effort by people like Mr. Stanton Musick of Wright Laboratory (MSOFE and MPLOT), Dr. E.C. Burt, Ms. Kim Corbitt, and Ms. Bea Kendziorzsky of MIT/Lincoln Laboratory (RCSToolLLBox), and Mr. Rob DeWall of Veda (signature processing software).

I must thank Mr. Michael Bryant and Mr. Rick Mitchell of the Wright Laboratory Target Recognition Technology Branch for suggesting high range resolution radar as an application for my proposals. I also thank Dr. Frank Kuhl of the U.S. Army, Dr. Fred Beckner of Cyberdynamics, and Mr. Harry Mieras of Raytheon for their support.

My wife and children have sacrificed as much as I have for this effort, and I thank them for the patience, love, and understanding that they have given me.

In closing, I offer my thanks to the Lord for helping me every day in so many ways, and for giving me the strength to finish this research.

Edmund W. Libby



AFIT/DS/ENG/93-05

APPLICATION OF SEQUENCE COMPARISON METHODS  
TO MULTISENSOR DATA FUSION  
AND TARGET RECOGNITION

DISSERTATION  
Edmund W. Libby  
LTC, USA

AFIT/DS/ENG/93-05

Approved for public release; distribution unlimited

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	


DTIC QUALITY INSPECTED 3


APPLICATION OF SEQUENCE COMPARISON METHODS  
TO MULTISENSOR DATA FUSION  
AND TARGET RECOGNITION

Edmund W. Libby, M.S.

LTC, USA

Approved:

<u>Peter S. Maybeck</u>	<u>21 July 93</u>
<u></u>	<u>21 July 93</u>
<u>Bruce W. Suter</u>	<u>21 July 93</u>



Robert A. Calico

Dean

## *Table of Contents*

	Page
Acknowledgements . . . . .	iii
List of Figures . . . . .	xiii
List of Symbols . . . . .	xvii
List of Abbreviations . . . . .	xviii
Abstract . . . . .	xx
 I. Research Objectives . . . . .	 1-1
1.1 Introduction . . . . .	1-1
1.2 Research Overview and Justification . . . . .	1-3
1.2.1 Object Recognition Scenarios of Interest . . . . .	1-6
1.2.2 Discussion: Sequences and Joint Likelihood . . . . .	1-8
1.2.3 One Approach . . . . .	1-10
1.2.4 Another Approach . . . . .	1-12
1.2.5 Development Plan . . . . .	1-12
1.3 Conclusion . . . . .	1-13
 II. Background . . . . .	 2-1
2.1 Introduction . . . . .	2-1
2.2 Pattern Recognition / Automatic Object Recognition . . . . .	2-1
2.2.1 Decision Theoretic Pattern Recognition . . . . .	2-2
2.2.2 Syntactic Pattern Recognition. . . . .	2-4
2.2.3 Example: Target Recognition by High Range Resolu- tion (HRR) Radar. . . . .	2-8
2.2.4 Pattern Recognition - Conclusion . . . . .	2-15

	Page
2.3 Object Tracking / State Estimation . . . . .	2-15
2.3.1 The Kalman Filter and Other Estimators . . . . .	2-16
2.3.2 Object Tracking with Kinematic Measurements Only.	2-29
2.3.3 Tracking with Kinematic and Signature Measurements.	2-34
2.3.4 Object Tracking / State Estimation – Conclusion . .	2-46
2.4 Sequence Comparison by Dynamic Programming . . . . .	2-46
2.4.1 Dynamic Programming. . . . .	2-47
2.4.2 Dynamic Time Warping for Speech Recognition. . .	2-49
2.4.3 General Sequence Comparisons. . . . .	2-54
2.4.4 The Larson and Peschon (L&P) Algorithm. . . . .	2-56
2.4.5 Relating Dynamic Time Warping (DTW) and the L&P Algorithm. . . . .	2-60
2.5 Dynamic Programming in Object Recognition . . . . .	2-62
2.5.1 Conception and Development of Motion Warping. .	2-62
2.5.2 Previous Developments by S. Kenyon. . . . .	2-64
2.5.3 Previous Developments by A. Amini, H. Yamada, <i>et al.</i>	2-64
2.5.4 Previous Developments by B. Burg, J. Gorman, <i>et al.</i>	2-64
2.5.5 Previous Developments by Y. Barniv <i>et al.</i> . . . . .	2-65
2.5.6 Previous Developments by Kramer and Demirbaş. .	2-66
2.5.7 Previous Developments by F. Le Chevalier <i>et al.</i> . .	2-67
2.5.8 Recent Developments by H. Mieras <i>et al.</i> . . . . .	2-71
2.5.9 Significance of the Le Chevalier and Mieras Efforts .	2-73
2.6 Multisensor Fusion . . . . .	2-74
2.6.1 General Overview . . . . .	2-74
2.6.2 Multisensor Fusion for Dynamic Objects . . . . .	2-76
2.7 Generalized Ambiguity Functions . . . . .	2-80
2.8 Conclusion . . . . .	2-86

	Page
III. Exploiting Joint Likelihood in Object Recognition . . . . .	3-1
3.1 Introduction . . . . .	3-1
3.2 Spaces and Dimensions. . . . .	3-3
3.2.1 Physical or Three-Space. . . . .	3-3
3.2.2 Aspect Angle Space. . . . .	3-3
3.2.3 Feature or Feature Observable Space. . . . .	3-4
3.2.4 Space of Allowable Warping Paths. . . . .	3-5
3.3 Joint Likelihood in Object Recognition . . . . .	3-5
3.4 Conventional Multiple Model Approaches for Dynamic Object Recognition . . . . .	3-9
3.4.1 Required Conditions and Estimator Alternatives . .	3-9
3.4.2 Kinematic/Aspect-Angle Estimators for Recognition	3-11
3.4.3 Summary . . . . .	3-12
3.5 Comparing Observed and Expected Feature Observables With- out Linear Prediction . . . . .	3-13
3.6 Applying the Larson and Peschon Approach in Recognition .	3-16
3.6.1 Facts and Assumptions . . . . .	3-17
3.6.2 Relating Path Probability $p(\mathbf{X}_{k,n}^a \mid \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ to Ob- ject Class Probability $p(\omega_i \mid \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ . . . . .	3-22
3.6.3 Relating $I^*(\mathbf{x}_k^a, k)$ to Object Class Probability $p(\omega_i \mid$ $\mathbf{Z}_k^f, \mathbf{Z}_m^d)$ . . . . .	3-27
3.6.4 Approximations for $p(\omega_i \mid \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ , and Associated Problems . . . . .	3-29
3.6.5 Obtaining "A Priori" Information for the L&P Ap- proach . . . . .	3-30
3.6.6 Issues in the Use of Larson and Peschon Methods in Object Recognition . . . . .	3-33
3.7 Classical Sequence Comparison in Object Recognition. . . . .	3-36

	Page
3.7.1 Motivating Classical Sequence Comparison in Object Recognition. . . . .	3-36
3.7.2 Implementing Classical Sequence Comparison in Object Recognition. . . . .	3-40
3.8 Sequence Comparison Concept Relationships . . . . .	3-45
3.8.1 Contrasting Prior Efforts with the Author's Research	3-47
3.8.2 Contrasting Classical Sequence Comparison-Based vs. Larson and Peschon-Based "Motion Warping" Concepts. . . . .	3-48
3.8.3 Comparing Use of Kinematic Information in the Author's Research with the Le Chevalier / Mieras Concepts . . . . .	3-51
3.8.4 Advantages of Predecessor Approaches . . . . .	3-52
3.9 A New Class of "Coupled" Estimators for Object Recognition	3-53
3.10 Assessing the Proposed Recognizers as Syntactic Approaches	3-55
3.11 Evaluating Algorithm Performance . . . . .	3-56
3.11.1 Introduction. . . . .	3-56
3.11.2 Conventional Performance Evaluation. . . . .	3-56
3.11.3 Performance Evaluation with the Generalized Ambiguity Function. . . . .	3-57
3.12 Chapter Summary . . . . .	3-60
IV. Residual Analysis for Sensor Fusion and Object Recognition . . . . .	4-1
4.1 Introduction . . . . .	4-1
4.2 A Tracking Filter for Target Recognition . . . . .	4-1
4.2.1 Design Philosophy . . . . .	4-1
4.2.2 The Filter State Model . . . . .	4-5
4.2.3 The Filter Measurement Model . . . . .	4-9
4.2.4 Filter Tuning . . . . .	4-13

	Page
4.3 Results and Discussion . . . . .	4-16
4.3.1 Scenarios 1A and 1B . . . . .	4-18
4.3.2 Scenario 2 . . . . .	4-27
4.4 Summary . . . . .	4-31
V. Dynamic Programming Sequence Comparison for Target Recognition . .	5-1
5.1 Introduction . . . . .	5-1
5.2 Concept Overview . . . . .	5-1
5.2.1 Tracking the Target . . . . .	5-1
5.2.2 Defining Likely Aspect Angle Regions and Paths . .	5-1
5.2.3 Matching Feature Observations to Aspect Angles . .	5-2
5.2.4 Evaluating Performance . . . . .	5-2
5.3 Generation of Simulated Target Signatures . . . . .	5-2
5.3.1 Introduction . . . . .	5-2
5.3.2 Signature Simulators: RCSToolLLBox and Others. .	5-4
5.3.3 RCSToolLLBox Models and Signatures . . . . .	5-6
5.3.4 Test Data Analysis for Noise Statistics. . . . .	5-12
5.3.5 Generating Simulated Signatures . . . . .	5-19
5.3.6 Summary . . . . .	5-25
5.4 Initial Concept Demonstration . . . . .	5-25
5.4.1 Test Objectives. . . . .	5-25
5.4.2 Test Procedures. . . . .	5-26
5.4.3 Test Results. . . . .	5-30
5.4.4 Conclusion. . . . .	5-31
5.5 Detailed Procedures for Motion Warping . . . . .	5-31
5.5.1 Introduction. . . . .	5-31
5.5.2 Tracking the Target . . . . .	5-32

	Page
5.5.3 Developing the Kinematic Aspect Angle Path and Error Bounds . . . . .	5-33
5.5.4 Extracting the Kinetically-Estimated Feature Observable Sequence. . . . .	5-45
5.5.5 Comparing the Measured Sequence to the Candidate Target Signatures. . . . .	5-45
5.5.6 Conclusion . . . . .	5-54
5.6 Research Results and Discussion . . . . .	5-54
5.6.1 Overview . . . . .	5-54
5.6.2 Kinematic State Estimation . . . . .	5-54
5.6.3 Recognition Algorithm Performance . . . . .	5-57
5.6.4 Effects of Considering Likely Kinematics: Restricting Transitions . . . . .	5-71
5.6.5 Miscellaneous Issues . . . . .	5-76
5.7 Summary . . . . .	5-78
 VI. Further Developments Exploiting Joint Likelihood in Object Recognition	 6-1
6.1 Introduction . . . . .	6-1
6.2 Sequence Comparison Methods for Single Object Identification	6-1
6.2.1 A New Class of Estimator for Object Recognition and Tracking. . . . .	6-1
6.2.2 Other Approaches for Object Recognition with Sensor Signature Sequences . . . . .	6-3
6.2.3 Sequence Comparison for <i>A Priori</i> -Likely Sequences. . . . .	6-3
6.3 Sequence Comparison Methods for Multi-Object Tracking and Data Association . . . . .	6-4
6.3.1 Motion Warping for Observation-to-Observation Association. . . . .	6-5
6.3.2 Joint Likelihood in Observation-to-Track Assignment. . . . .	6-6
6.4 Mathematical Issues in Multisensor Fusion for Recognition . . . . .	6-6



	Page
6.4.1 Parameter Estimation and Object Recognition . . .	6-7
6.4.2 The Motion Warping Process as a Nonlinear Functional.	6-8
6.4.3 Sequence Comparison for Multiresolution Analysis. .	6-8
6.5 Alternative Approaches for Sequence Comparison. . . . .	6-9
6.6 Miscellaneous Issues and Extensions To This Research . . . .	6-9
6.6.1 Feature Space and Distance Metric Choices. . . . .	6-9
6.6.2 Application to Other Classes of Objects. . . . .	6-10
6.6.3 Path Discretization. . . . .	6-10
6.6.4 Alternative Kinematic/Aspect-Angle Filter Designs	6-11
6.6.5 Application of the Larson and Peschon Equations in Speech Recognition and Other Areas . . . . .	6-11
6.7 Summary. . . . .	6-12
VII. Summary and Conclusion . . . . .	7-1
Appendix A. Definitions . . . . .	A-1
Appendix B. Background Information on Pattern Recognition . . . . .	B-1
B.1 Taxonomy of Pattern Recognition Concepts for 3-D Objects.	B-1
B.2 Decision Theoretic Object Recognition. . . . .	B-4
B.2.1 Decision Theoretic Methods – Survey. . . . .	B-4
B.2.2 Statistical Library Approaches. . . . .	B-6
B.2.3 Model-Based Approaches. . . . .	B-8
B.3 Pattern Recognition Using Global Descriptors. . . . .	B-9
B.3.1 Moments. . . . .	B-9
B.3.2 Fourier Descriptors. . . . .	B-10
B.3.3 Miscellaneous Global Descriptors. . . . .	B-10
B.4 Pattern Recognition Using Correspondences. . . . .	B-10
B.5 Pattern Recognition Using Correlations. . . . .	B-11
B.6 Summary . . . . .	B-11

	Page
Appendix C. Detailed Equations for Estimation and Tracking Algorithms	C-1
C.1 Introduction . . . . .	C-1
C.2 The Kendrick / Maybeck / Reid Estimator . . . . .	C-1
C.2.1 Kinematic Filter . . . . .	C-1
C.2.2 Aspect Angle Filter. . . . .	C-4
C.3 The Andrisani / Kuhl / Gleason Estimator . . . . .	C-6
C.3.1 State Equations. . . . .	C-6
C.3.2 Measurement Equations. . . . .	C-9
C.4 The Swarder / Hutchins Estimator . . . . .	C-10
C.5 Nonlinear Kinematic Tracker Effort by Bishop. . . . .	C-12
C.6 Equations and Additional Results for Optimal Smoothing . .	C-13
C.6.1 Overview . . . . .	C-13
C.6.2 The Fixed Point Smoother . . . . .	C-14
C.6.3 The Fixed Lag Smoother . . . . .	C-16
C.6.4 Optimal Smoothing with MSOFE Output . . . . .	C-17
C.6.5 Curve Fitting . . . . .	C-19
C.6.6 Results and Discussion . . . . .	C-20
C.7 Equations for Direction Cosine Matrix (DCM) Generation . .	C-26
C.7.1 DCM for Sensor-to-Inertial Frame Transformation .	C-26
C.7.2 DCM for Target Lift-to-Inertial Frame Transformation	C-30
Vita . . . . .	VITA-1
Bibliography . . . . .	BIB-1

## *List of Figures*

Figure	Page
1.1. Tank Target – True vs. Estimated Trajectory (Notional) . . . . .	1-7
1.2. The Hypothetical Aspect Angle Sphere . . . . .	1-9
2.1. Typical High Range Resolution Radar Signatures – Rear Aspect . . . . .	2-10
2.2. Multiple Model Estimation Algorithm . . . . .	2-26
2.3. The Singer Model for Acceleration Probability Density . . . . .	2-32
2.4. The Kendrick-Type Kinematic/Aspect-Angle State Estimator . . . . .	2-37
2.5. The Andrisani-Type Kinematic/Aspect-Angle State Estimator . . . . .	2-39
2.6. Aspect Angle Information Fusion in the Kendrick-Type Estimators . . . . .	2-41
2.7. A Minimum Cost Path Defined By Dynamic Programming . . . . .	2-48
2.8. Classical Dynamic Time Warping . . . . .	2-51
2.9. Motion Warping in a Two-Dimensional Aspect Angle Region . . . . .	2-55
2.10. Ambiguity Functions for Estimators of Varied Quality . . . . .	2-83
3.1. A Measurement Function Unsuitable for Conventional Filtering . . . . .	3-9
3.2. Aspect Angle Windows and the Nominal Aspect Angle Path . . . . .	3-18
3.3. Typical Matching Paths – Larson and Peschon Algorithm . . . . .	3-24
3.4. “Overhead View” – Larson and Peschon Path . . . . .	3-25
3.5. Two Aspect Angle Paths . . . . .	3-34
3.6. Motion Warping – Aspect Angle Paths and Multiple Models . . . . .	3-37
3.7. Correspondence Between Kinematically Estimated and True Aspect Angle Paths . . . . .	3-39
3.8. Typical Matching Paths – Classical Sequence Comparison . . . . .	3-41
3.9. Kinematic-to-True Aspect Angle Association Improvement With Warping . . . . .	3-44
3.10. Relating Earlier Developments to the Author’s Research . . . . .	3-46
3.11. Typical Matching Path – Fixed Bound Algorithm . . . . .	3-49

Figure	Page
4.1. Tracking / Recognition Scenario for Discussion . . . . .	4-3
4.2. Distinction Between $\alpha_b$ and $\alpha_c$ . . . . .	4-6
4.3. Z (Vertical) Sensor Axis Position Residuals, Correct Model, Scenario 1A	4-20
4.4. Z (Vertical) Sensor Axis Position Residuals, Wrong Model, Scenario 1A	4-20
4.5. $k_{aero}$ Error, Correct Model, Scenario 1A . . . . .	4-21
4.6. $k_{aero}$ Error, Wrong Model, Scenario 1A . . . . .	4-21
4.7. Residual / State Log Likelihoods, Correct Model, Scenario 1A . . . . .	4-22
4.8. Residual / State Log Likelihoods, Wrong Model, Scenario 1A . . . . .	4-22
4.9. Residual / State Log Likelihoods, Correct Model, Scenario 1A, Moving Window Sum (one second, ten measurements) . . . . .	4-24
4.10. Residual / State Log Likelihoods, Wrong Model, Scenario 1A, Moving Window Sum (one second, ten measurements) . . . . .	4-24
4.11. Z (Vertical) Sensor Axis Position Residuals, Correct Model, Scenario 1B	4-25
4.12. Z (Vertical) Sensor Axis Position Residuals, Wrong Model, Scenario 1B	4-25
4.13. $k_{aero}$ Error, Correct Model, Scenario 1B . . . . .	4-26
4.14. $k_{aero}$ Error, Wrong Model, Scenario 1B . . . . .	4-26
4.15. $k_{aero}$ Error, Correct Model, Scenario 2 . . . . .	4-28
4.16. $k_{aero}$ Error, Wrong Model, Scenario 2 . . . . .	4-28
4.17. Residual / State Log Likelihoods, Correct Model, Scenario 2 . . . . .	4-29
4.18. Residual / State Log Likelihoods, Wrong Model, Scenario 2 . . . . .	4-29
4.19. Residual / State Log Likelihoods, Correct Model, Scenario 2, Moving Window Sum (one second, ten measurements) . . . . .	4-30
4.20. Residual / State Log Likelihoods, Wrong Model, Scenario 2, Moving Win- dow Sum (one second, ten measurements) . . . . .	4-30
5.1. The SRCRCS/RCSToolLLBox Radar Target Coordinate Frame . . . . .	5-7
5.2. Boeing B-737 . . . . .	5-9
5.3. Yakovlev YAK-28 . . . . .	5-9
5.4. Mikoyan MIG-21 . . . . .	5-10

Figure	Page
5.5. Sukhoi SU-22 . . . . .	5-10
5.6. McDonnell-Douglas F-4 . . . . .	5-11
5.7. McDonnell-Douglas F-4 with stores (bombs) . . . . .	5-11
5.8. Mikoyan MIG-21 with Extra Point Scatterers . . . . .	5-12
5.9. F-4, 50% Morph, and MIG-21 . . . . .	5-13
5.10. Measured Signatures: Means and Variances – Rear Aspect . . . . .	5-16
5.11. Measured Signatures: Means and Variances – Front/Side Aspect . . . . .	5-16
5.12. Measured Signatures: Means and Variances – Front Aspect . . . . .	5-17
5.13. Simulated Signatures: RCSToolLLBox-Generated and Noise-Augmented . . . . .	5-20
5.14. Simulated Signatures: RCSToolLLBox-Generated and Noise-Augmented . . . . .	5-20
5.15. Simulated Signatures: RCSToolLLBox-Generated and Noise-Augmented . . . . .	5-21
5.16. Simulated Signatures: RCSToolLLBox-Generated and Noise-Augmented . . . . .	5-21
5.17. Simulated Signatures: RCSToolLLBox-Generated and Noise-Augmented . . . . .	5-23
5.18. Simulated Signatures: RCSToolLLBox-Generated and Noise-Augmented . . . . .	5-23
5.19. Simulated Signatures: RCSToolLLBox-Generated and Noise-Augmented . . . . .	5-24
5.20. Typical Range Sweep Sequence – A . . . . .	5-28
5.21. Typical Range Sweep Sequence – B . . . . .	5-28
5.22. Typical Range Sweep Sequence – C . . . . .	5-29
5.23. Coordinate Frame Definitions . . . . .	5-35
5.24. Accelerations for Coordinated Turns . . . . .	5-36
5.25. Violation of the Principle of Optimality From Local Path Normalization . . . . .	5-51
5.26. Acceleration Estimation Error: Mean +/- One Standard Deviation . . . . .	5-55
5.27. Generalized Ambiguity Function for Case 1 . . . . .	5-58
5.28. Generalized Ambiguity Function for Case 2 . . . . .	5-59
5.29. Percent Correct Recognition for Case 3 . . . . .	5-61
5.30. Generalized Ambiguity Function for Case 3 . . . . .	5-63
5.31. Generalized Ambiguity Function for Case 3 (Modified) . . . . .	5-64

Figure	Page
5.32. Generalized Ambiguity Function for Case 4 . . . . .	5-65
5.33. Generalized Ambiguity Function for Case 5 . . . . .	5-66
5.34. Percent Correct Recognition for Case 6 . . . . .	5-68
5.35. Generalized Ambiguity Function for Case 6 . . . . .	5-69
5.36. ML Aspect Angle Estimates on the Wrong Target . . . . .	5-72
5.37. ML Aspect Angle Estimates on the Correct Target . . . . .	5-75
6.1. Problem: Associating Signatures From Imaging and Radar Sensors . . .	6-5
C.1. Target Trajectories for Smoother Discussion . . . . .	C-21
C.2. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 2 . .	C-23
C.3. Error in Target Position Estimates – Inertial Y Axis, Scenario 2 . . . .	C-23
C.4. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 3 . .	C-24
C.5. Error in Target Position Estimates – Inertial Y Axis, Scenario 3 . . . .	C-24
C.6. Error in Target Acceleration Estimates – Inertial Z Axis, Scenario 3 . .	C-25
C.7. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 4 . .	C-25
C.8. Error in Target Acceleration Estimates – Inertial Z Axis, Scenario 4 . .	C-27
C.9. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 5 . .	C-27
C.10. Error in Target Acceleration Estimates – Inertial Z Axis, Scenario 5 . .	C-28
C.11. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 5 . .	C-28

# *List of Symbols*

Symbol	Page
<i>a posteriori</i> probability: $p(\omega_i   \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ . . . . .	2-77
<i>a priori</i> object class probability: $p(\omega_i)$ . . . . .	2-76
aspect angle state transition likelihood: $p(\mathbf{x}_{j+1,n}^a   \mathbf{x}_{j,n}^a, \mathbf{Z}_m^d, \omega_i)$ . . . . .	3-21
aspect angle state value: $\mathbf{x}^a$ . . . . .	3-4
covariance of measurement error: $\mathbf{R}(t_i)$ . . . . .	2-17
dynamics driving noise strength: $\mathbf{Q}(t)$ . . . . .	2-17
estimated error covariance after measurement update at time $t_i$ : $\mathbf{P}(t_i^+)$ . .	2-18
feature or signature measurements: $\mathbf{Z}_k^f = \{z_1^f, z_2^f, z_3^f, \dots, z_k^f\}$ . . . . .	2-76
filter residual: $[\mathbf{z}_i - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)]$ . . . . .	2-18
generalized ambiguity function: $\mathcal{A}_k(\Omega, \Omega_i)$ . . . . .	2-81
generalized likelihood function: $L[\mathbf{x}, \mathbf{y}, \mathbf{Z}_k]$ . . . . .	2-77
$i$ -th object class: $\omega_i$ . . . . .	2-2
Kalman filter gain at time $t_i$ : $\mathbf{K}(t_i)$ . . . . .	2-18
Larson and Peschon conditional probability index: $I^*(\mathbf{x}_k^a, k)$ . . . . .	3-27
"Larson and Peschon" estimate of the aspect angle path: $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ . . . . .	3-20
$m$ kinematic measurements: $\mathbf{Z}_m^d = \{z_1^d, z_2^d, z_3^d, \dots, z_m^d\}$ . . . . .	2-77
measurement matrix at time $t_i$ : $\mathbf{H}(t_i)$ . . . . .	2-18
number of possible aspect angle paths: $N_p$ . . . . .	3-19
one estimated or predicted feature observable value: $\hat{\mathbf{z}}_i^f = \mathbf{h}[\hat{\mathbf{x}}^a(t_i)]$ . . . .	3-38
probability of starting cell location: $p(\mathbf{x}_{0,n}^a   \mathbf{Z}_m^d, \omega_i)$ . . . . .	3-21
residual covariance: $\mathbf{A}_i(t_k) = \mathbf{H}(t_k)\mathbf{P}(t_k^-)\mathbf{H}^T(t_k) + \mathbf{R}(t_k)$ . . . . .	2-24
sequence of estimated or predicted feature observable values: $\hat{\mathbf{Z}}^f = \mathcal{H}[\hat{\mathbf{x}}^a(t)]$ . .	3-38
set of all aspect angle paths: $\mathcal{X}_{pk}^a$ . . . . .	3-19
(super-) set of aspect angle states: $\mathcal{X}^a$ . . . . .	3-19

### *List of Abbreviations*

Abbreviation	Page
AOR (Automatic Object Recognition) . . . . .	1-1
BRL (Ballistics Research Laboratory) . . . . .	5-6
BRL-CAD (BRL Computer Automated Design Package) . . . . .	5-5
CRLB (Cramér-Rao lower bound) . . . . .	2-85
DCM (direction cosine matrix) . . . . .	C-26
DOF (degrees of freedom) . . . . .	3-3
DP (dynamic programming) . . . . .	2-47
DTW (Dynamic Time Warping) . . . . .	2-49
FB (Fixed Bound Algorithm) . . . . .	5-47
F-4 (U.S. multirole tactical fighter) . . . . .	5-12
GAF (generalized ambiguity functions) . . . . .	2-80
GD (General Dynamics) . . . . .	2-11
HMM (hidden Markov model) . . . . .	6-9
HRR (High Range Resolution (Radar)) . . . . .	2-8
IL (Independent Look Algorithm) . . . . .	5-46
ISAR (Inverse Synthetic Aperture Radar) . . . . .	2-11
L&P (Larson and Peschon) . . . . .	2-56
L&P (Larson and Peschon Algorithm For Target Recognition) . . . . .	5-47
MAP (maximum <i>a posteriori</i> ) . . . . .	2-56
MFML (Minimum Feature, Maximum Likelihood (Metric)) . . . . .	2-13
MIG-21 (Russian interceptor) . . . . .	5-8
ML (maximum likelihood) . . . . .	2-61
MMAE (multiple model adaptive estimator) . . . . .	2-23
MPLOT (plot postprocessor for MSOFE) . . . . .	5-54
MSOFE (Multimode Simulation for Optimal Filter Evaluation) . . . . .	4-16



Abbreviation	Page
NCTR (Non-Cooperative Target Recognition) . . . . .	2-1
PKA (Perfect Knowledge of Aspect Algorithm) . . . . .	5-47
PSRI (position-, scale-, and (in plane) rotation-invariant) . . . . .	1-5
RCSToolLLBox . . . . .	5-4
SCAMP . . . . .	5-6
SRCRCS . . . . .	5-4
sup (supremum) . . . . .	A-11
XPATCH . . . . .	5-5
1-D Warp (One Dimensional Classical Sequence Comparison Algorithm) .	5-48
2-D Warp (Two Dimensional Classical Sequence Comparison Algorithm) .	5-53

*Abstract*

This research addresses methods for exploiting the spatio-temporal joint likelihood of observed kinematic and nonkinematic (sensor signature) physical events to improve dynamic object and target recognition. A principal direction is the application of dynamic programming sequence comparison techniques to condition matching of object signatures to known models *according to observed kinematics* – that is, to use information from observed kinematics in determining allowable aspect angles with which observed signatures may be matched on models for candidate objects. A second direction is the application of kinematic/aspect-angle Kalman filter trackers to condition kinematic tracking according to observed signatures. These conditioning processes dramatically reduce ambiguity in object recognition, and can be used together or separately to allow computation of *a posteriori* probabilities of object class membership using Bayesian methods. Proposals are supported by results of simulated target tracking and high range resolution radar signature analysis. The original contributions of this effort include: (1) new approaches for and theoretical understanding of syntactic methods in multisensor fusion and dynamic object recognition; (2) extension of estimation and tracking techniques to allow object recognition; and (3) introduction of a new performance evaluation technique and approach for establishing performance bounds in dynamic object and target recognition.

# APPLICATION OF SEQUENCE COMPARISON METHODS TO MULTISENSOR DATA FUSION AND TARGET RECOGNITION

## I. Research Objectives

### 1.1 Introduction

The last ten years have seen vast effort applied to the problem of fusing information from multiple sensors for accurate object tracking and, ideally, automatic object recognition (AOR) [8, 28, 40, 122, 218]<sup>1</sup>. Acceptable solutions exist for many of these and other problems posed under the heading of *multisensor data fusion*, but practical automatic object recognition is not yet a reality. The fundamental shortcomings in this area are reflected by the variety of efforts underway to provide better results [44, 97, 124, 126, 224]. One aspect of sensor fusion which has received comparatively little attention is the fusion of information on object *motion* with information on other observable object characteristics – *feature observables* or *sensor signatures* [120, 121, 4, 122, 209].

The intent of this research has been to define new approaches for fusing sensor information to improve recognition of dynamic objects in general and tactical targets in particular. The common link between the resulting approaches is their exploitation of the characteristic relationships between observable motion and sensor signatures for typical object classes and behaviors of interest. Inherently, these relationships are forced by physics, and involve the concept of *joint likelihood* of object motion and other observables – that is, when the correct object model is associated with measurements from an unclassified object, the development of measurements and object state variable estimates over time should tend to be consistent *in all observable domains*. An incorrect object class association, on the other hand, is more likely to betray itself through some inconsistency between the expected and observed motion and/or object signatures.

---

<sup>1</sup>Note: throughout this document, citation listings appearing out of numerical order indicate that the citations are listed in order of decreasing relevance.

Early in this research, it was observed that many of the problems inherent in melding information from position/motion sensors and feature sensors could perhaps be addressed using techniques historically applied in speech processing and related sequence comparison efforts [176, 193, 182, 195]. Another observation behind this effort is that previously proposed target tracking algorithms [4, 120, 121, 122, 209] would fail in obvious ways when incorrect assumptions about target class were made, and that a target recognition algorithm could use that failure as evidence of an incorrect choice of target class.

Ultimately driven by these insights, the original contributions of this research are the following:

- (1) The extension of previous work in multisensor target *tracking* [4, 120, 121, 122, 209] to allow object *recognition* through the method of multiple model estimator residual analysis.
- (2) The application of classical sequence comparison techniques, as used in speech processing, chromosome comparison, and other areas, to multisensor fusion for dynamic object recognition.
- (3) The association of (a) dynamic programming-based state estimation techniques with (b) classical sequence comparison techniques, and application of these state estimation techniques to multisensor fusion for dynamic object recognition.
- (4) The joining of (1), (2), and (3) to create a new estimator structure exploiting *joint likelihood* of object motion and sensor signature measurements for object recognition.
- (5) Contributions to the theory of dynamic object recognition as a problem in *syntactic* pattern recognition and joint likelihood of all observed events.
- (6) The application of classical Bayesian parameter estimation and generalized ambiguity function techniques for multisensor object recognition.

These contributions provide new approaches for combining information to make object recognition decisions, and new or fresh understanding of previous efforts by other researchers. In most cases, the information to be fused is already available from common state-of-the-art sensors and can be integrated using techniques to be shown herein with the addition of computational power only – no new sensors are required. In particular, unlike neural nets, hidden Markov models, and other currently popular information fu-

sion algorithms [169, 81, 174, 175, 68], the approaches discussed here do not establish decision-making parameters through training processes over which the user has no direct visibility – all decisions are easily traced and accountable according to Bayesian theory, dynamic programming, and classic parameter estimation methods. Moreover, all state variables employed in this effort correspond to well-understood physical processes or statistical representations for those processes.

This dissertation is organized into seven chapters and three supporting appendices. This first chapter outlines and justifies the research. Chapter II discusses the state of the art in pattern recognition, target tracking, and sensor fusion. Chapter III exploits the material of Chapter II to define a new class of object recognition algorithms. Chapter IV discusses one element in this class, the extension of kinematic/aspect-angle tracking filters and residual/state monitoring to multisensor object recognition. Chapter V discusses another element in this class, the application of dynamic programming-based sequence comparison techniques to multisensor object recognition. Chapter VI recommends extensions to this effort to explore issues and options not addressed here. The concluding Chapter VII reviews the major points of this effort. Appendix A gives a glossary of key terms. Appendix B provides fundamental background material in pattern recognition. Appendix C lists particular equations and provides results which are considered too involved for the body of the text or are not part of the original effort of this research.

## *1.2 Research Overview and Justification*

Contemporary researchers generally classify the information available from sensors into two broad categories – kinematic (or spatial) and nonkinematic (or nonspatial) [8:297–302] [218:165]. Kinematic or motion-related information is characteristically limited to measurements of object centroid position and velocity. Nonkinematic information is a much more diverse categorization, classically including “feature attributes” such as radar signature, optically-derived shape descriptors, the presence or absence of certain forms of electromagnetic emissions, and so on. Nonkinematic information is often a direct function of the object or target aspect angle relative to the sensor, reflecting the “appearance” of that object to the sensor in the appropriate spectral or algorithmic sense. As we shall

see, the dependence of aspect angle on kinematics makes the term "nonkinematic information" something of a misnomer – the author chooses to refer to these quantities as *feature observables* or *sensor signatures*, as noted in the chapter introduction.

In general, schemes for fusing kinematic information involve versions of the Kalman filter/state estimator, in which measured values are compared to mathematical models of object behavior to derive kinematic estimates that are optimal in some sense. This field is well-developed and diverse, with a rich body of supporting literature developed over the past 40 years [153, 154, 10].

In contrast, the fusion of nonkinematic information for recognition and tracking is a much newer field. Many approaches to this problem have been proposed [218:213–261], but the most popular alternatives involve only two basic approaches. First, the "decision function" or "nearest neighbor" approach [212:13], involves some form of distance measure between a vector of measured values for selected features derived from an unclassified object on one hand, and either (1) sets or *clusters* of such feature values derived from *a priori* testing of known object classes (for object recognition) [28, 227], or (2) clusters representing current tracks in a trackfile (for observation-to-track assignment) [8:312].

The second, or probabilistic/statistical approach, is based on maximum likelihood (or *classical inference*) methods [218:216–219]), Bayesian probability [212] [218:220–222], or Dempster-Shafer evidence accrual [33:380–386]. Here the user makes judgments regarding the probability that his measurements could occur, conditioned on the presence of one member of a set of certain likely object types, and combines these judgments in the appropriate framework to postulate which object type generated the observed data [14] [8:199–209]. It should be noted that the decision function and statistical approaches are intimately related in both theory and practice [212, 92].

The object databases against which nonkinematic features have been classically compared have been just that – tabular data arrays, or feature space mappings, or decision surfaces in a feature space of one form or another. Increasingly, however, researchers are turning to model-based approaches, in which the feature values can be derived on-line in some fashion from a three-dimensional representation of the object [169:111–139]. In ei-

ther case, often these features are position-, scale-, and (in-plane) rotation-invariant (PSRI) *global descriptors* or similar quantities, generally functions of the entire object representation at a given aspect angle, such as Hu moments and Fourier descriptors [73, 111, 217] (discussed in App. B).

The problem with this taxonomy, however, is that it is misleading to impose a definitive distinction between "kinematic" and "nonkinematic" forms of information. With few exceptions, nonkinematic features are functions of the object-sensor aspect angle (and other factors, of course, like weather). But this aspect angle is immediately a function of both object and sensor (ownership) kinematics. An extensive review of the literature shows that this connection, although recognized by some [4, 120, 121, 122, 209], has not generally been exploited. It is apparent to the author and others that sensor fusion efforts do not generally use information about motion which can be extracted from observed features, or information about features which can be extracted from motion [9:177-178] [30].

In the field of classical pattern recognition, considerable effort has been made to solve the problem of "recognition, tracking, and pose estimation of arbitrarily-shaped 3-d objects" (where the term "pose" refers to the object's aspect angle relative to the sensor) [102, 105, 110, 179]. These include a number of techniques to extract shape or structure from motion [110:400-422] [117, 116, 206]. Generally these techniques depend on (1) making correct assignments between points (or loci of points) on the observed object and corresponding points (or loci) on a model representation [69, 139], or (2) matching observed values of some global descriptor with model- or database-derived values [47, 73, 111]. Techniques in the first category usually depend on high quality images of the observed object, and are very sensitive to performance degradation due to the quality of images available from typical remote imaging sensors. Techniques in the second category are sensitive to errors in segmentation (separating the object from its background), occlusion (the presence of other objects blocking the object of interest-to-sensor line of sight), and so on. There are as yet no final "best" answers.

It seems clear that additional research is warranted to find methods for fusing data from "kinematic" and "nonkinematic" sensors. Note that henceforth, because of the mis-

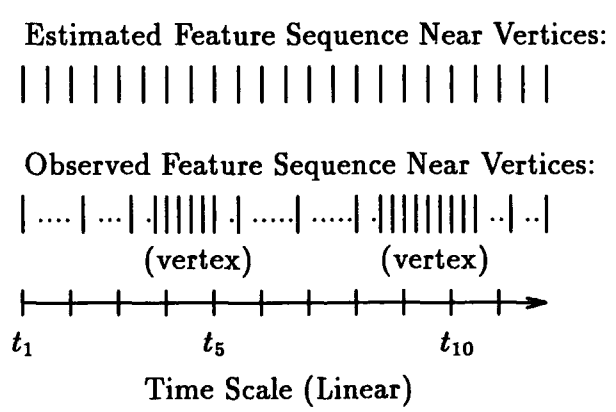
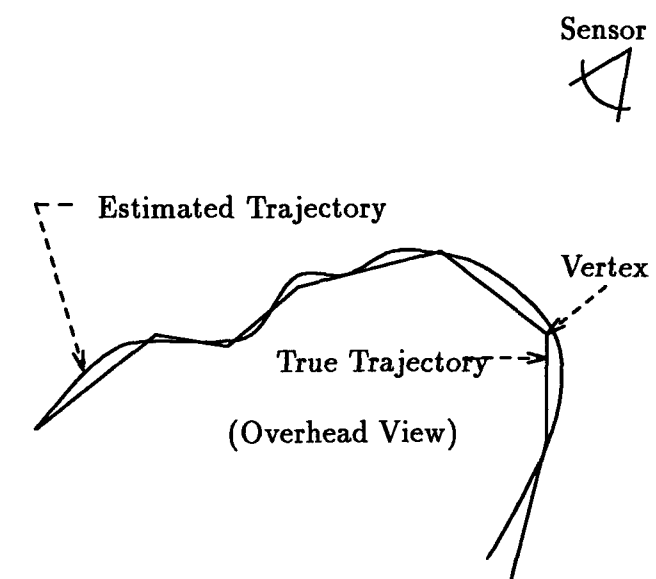
leading connotation of the term "nonkinematic," we will refer to "nonkinematic" quantities as "feature observables" or "sensor signatures."

*1.2.1 Object Recognition Scenarios of Interest.* Two promising military applications of kinematic-feature observable fusion for automatic object recognition would appear to be: (1) fusion of infrared image-derived angle, angle rate, and feature observable measurements with laser ranger measurements, and (2) fusion of high range resolution (HRR) radar "range sweep" waveform returns and centroid kinematic data from the same radar system (the latter an example of *multisensor* fusion in the sense that the different items of information are extracted by distinctly different sensors within the radar system). Note the relationship between true and estimated kinematics and signatures for targets in these scenarios:

Case 1: Consider the process of tracking a main battle tank of unknown class in a planar turn. Unlike wheeled vehicles, conventional tanks (more generally, vehicles propelled by tracks, rather than wheels) do not turn with a constant or even continuous radius of curvature, and in fact their motion can be described as nearly piecewise linear [86]. Our sampled-data sensor provides a sequence of signature vectors (e.g., classical global descriptors like Hu moments and Fourier descriptors), as measured at discrete times over the observation period. Concurrently, from conventional range (laser ranger) / angle tracking and kinematic state estimation alone, for any feasible target class, we can hypothesize a sequence of expected signature vectors. Comparing the observed sequence to the kinematically-estimated sequence for the *correct* target model in Fig. 1.1, we see that their differences can be described fundamentally in terms of expansions and contractions of one sequence relative to the other. Is it possible to compare these two sequences such that their origin target classes can be seen to be identical, despite expansions and contractions? Will a *sequence comparison* process reduce the likelihood of incorrect identifications, in comparison with other approaches?

Case 2: We sense a moving aircraft with a high range resolution (HRR) radar, from which we obtain sampled-data measurements of target position, doppler velocity (see definition in App. A) and "range sweeps" – measurements of radar cross section in relatively





Note: Contractions of Observed Sequence  
Relative to Estimated Sequence are due to  
higher than expected turn (angular rotation)  
rates at these times.

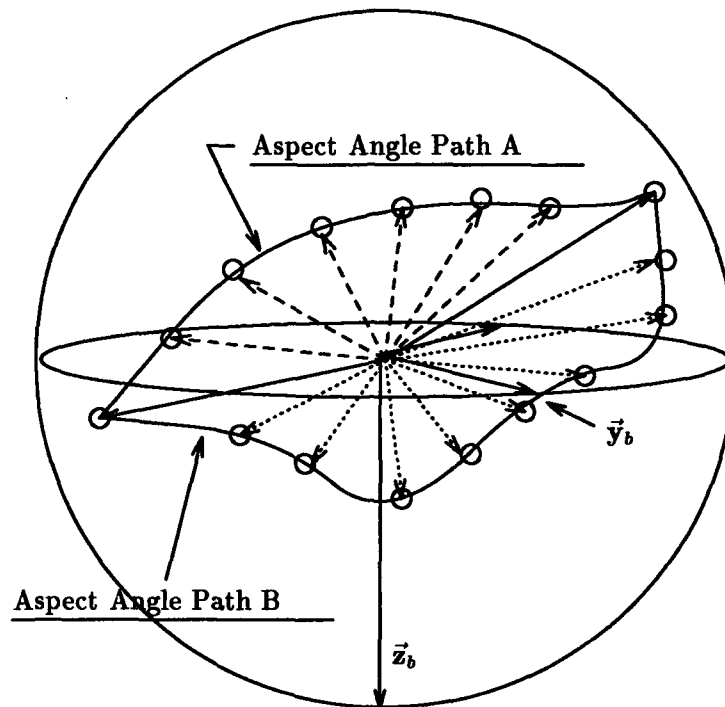
Figure 1.1. Tank Target – True vs. Estimated Trajectory (Notional)

fine (small with respect to typical target dimensions) range increments or “bins” along the target-sensor axis. Stochastic phase interactions between returns from scatterers in any given range bin make the range sweeps quite noisy, causing deviations from model and test predictions and limiting our ability to judge the class of the target from any one or perhaps several returns. Generally, however, distinctive elements (peaks) of the range sweeps stand out when returns are averaged, or viewed over time as the aspect angle changes, producing a distinctive sequence like the “sinogram” noted in [81] (for simulation-derived examples, the reader may wish to look ahead to Figs. 5.20, 5.21, and 5.22). As in Case 1, the radar position and velocity measurements allow us to estimate the motion of the target centroid, from which we can estimate the target-sensor aspect angle based on the flight control physics required to achieve that motion. Thus, as in the previous case, for any candidate target class, we can hypothesize a sequence of *expected* signatures for comparison with the sequence of *observed* signatures. The same questions apply as in Case 1 – the object of this research was to answer those questions.

*1.2.2 Discussion: Sequences and Joint Likelihood.* In both cases above, we wish to assign an object (target) to one class from a set of classes represented by models which give feature observable values as a function of aspect angle, and define the kinematic behavior of each object class. From our knowledge of the target (and ownship) kinematics, and the dynamic limitations for each class, we can estimate the target-sensor aspect angle as the target executes a maneuver. Now suppose that, centered on the origin of the classical three-axis target body coordinate frame (“pitch, roll, and yaw” axes, using air vehicle terminology), we hypothesize a 3-D sphere of unit radius, fixed with respect to that body frame.

Fig. 1.2 illustrates the sphere and body frame axes – see Fig. 5.23 for the classical relationship between body frame axes and structure for an aircraft target. Throughout this dissertation, this “*hypothetical aspect angle sphere*” will be a key framework for discussion.

As the target-sensor aspect changes, the target-sensor unit vector traces an *aspect angle path* or aspect angle “track” like path “A” or path “B” on the target sphere in Fig. 1.2. At any point on the target sphere, and therefore at any point along an aspect angle path on



Note:  $\vec{x}_b$  lies at rear of "equatorial plane."

Figure 1.2. The Hypothetical Aspect Angle Sphere

some model, we can predict a value for the corresponding feature observables. Recording these predictions at discrete points in time along the kinematically-estimated aspect track, we obtain a one-dimensional string or sequence of discrete values. If our model choice and aspect angle estimates are correct, and all other relevant factors are equal between measurement and prediction, this predicted sequence should correspond exactly to the sequence of measured values.

Of course, for a variety of reasons, the predicted and measured sequences will never correspond exactly, even where the model corresponds exactly to the observed target. For example, variations in the signature generation process, atmospheric transmission, and sensor processing errors will induce noise in our measurements that cannot be predicted by a finite-dimensional mathematical model. More subtly, the deviation of true target kinematics from model assumptions will cause the true aspect angle path to lie off of

the kinematically-estimated or "nominal" aspect angle path. This could happen for an aircraft target, for example, if our model assumes conventional *coordinated turn* dynamics, but the target flies using advanced control-configured dynamics [112] that do not follow coordinated turn rules.

We may note also that for some choices of object or target, feature space and comparison metric (between observed and predicted feature values), each individual element in the observed feature sequence could be used to define a "maximum likelihood" *pose* or aspect angle estimate. If this pose information is known to be accurate with respect to the correct object or target class, it may be usable with the target kinematic information to improve estimates for the *future* kinematics and aspect angle / feature sequence.

In either case, it is clear that, for the proper association of an unclassified target or other dynamic object with the correct model from a set of known classes, we expect that the *joint likelihood* of the observed quantities in kinematic, feature observable, and aspect angle domains will be consistent *over time* – an incorrect association will be less likely to exhibit the correct combination of behavior. We now consider two basic approaches for exploiting this expectation of high joint likelihood. In both cases, treating the observable events as sequences over time will be critical, since sequences of observations from physical objects inherently contain information about the joint likelihood of deriving those observations from any particular class of objects.

*1.2.3 One Approach.* One approach for fusing kinematic and feature observable data is suggested by the speech recognition technique called "dynamic time warping" (DTW) [176, 182, 193], one of a class of techniques for sequence comparison [195] which employ dynamic programming (DP) [23, 71]. DTW is a method for comparing sequences or *strings* of feature vectors or functions extracted from discretized speech against a "library" of feature vector strings corresponding to selected words. A close match between an observed or measured vector string and a library vector string establishes the presence of that word, or a sound sequence "close" in some sense to the matched library word.

The closeness of two vector sequences is established by a dynamic programming-based process, which "warps" each library sequence in a particular fashion to make it

resemble the observed sequence (or vice-versa), and establishes a warping path distance or "cost" for the matching process. This cost is often simply the sum of the individual costs, or dissimilarity measures, obtained by the matching of one or more elements of one sequence to one or more elements of the other sequence. The library vector sequence with the minimum warping path cost "wins" the comparison process. Note that the warping is required even when the observed vector string and the library vector string correspond to the same word, due to variations in factors like speaker pronunciation and background noise.

The applicability of a DP sequence comparison process to sensor fusion arises from the fact that sequences develop naturally in object recognition – observations seldom occur only *once* in a given scenario. As we saw in the previous section, one often measures a sequence of feature observable vectors or functions (e.g., "range sweeps" from high range resolution radar) as the object-sensor aspect angle changes, while from the motion of the object, one can estimate the object-sensor aspect angle for each potential object class as the features were observed [5, 77, 120, 121]. Given the aspect angle estimate sequence for each class, we can then generate a sequence of library vectors or functions representing the feature observable sequence which *should* have been produced by each model class while executing the observed maneuvers.

The basic similarity of these vectors or functions to speech data is immediately evident, and suggests application of a DP-based matching technique. The matching will never be exact, yet we may be able to achieve an association between observed and expected sequences which compensates for errors (noise) in the kinematic estimate, feature observable measurements, and model, to give a better model-object association than by comparing kinematics or features in isolation, or by comparison without "warping" (a linear comparison). A particularly attractive aspect of the proposed fusion scheme is that it has the potential to work for any object with (1) feature observables that can be expressed as functions of aspect angle relative to the object body frame, and (2) dynamics that are (a) restricted by orientation and (b) can be modelled for state estimation.

Dynamic programming sequence comparison-based fusion processes will henceforth be referred to as *motion warping* in this effort. We will see, however, that dynamic pro-

gramming algorithms not heretofore associated with classical sequence "warping" are also eminently applicable for our purpose.

*1.2.4 Another Approach.* A number of techniques have been proposed which estimate pose (aspect) directly from feature observables [102, 105, 208], requiring no input of kinematic information. These estimates of pose can in turn form the basis for a target tracker using kinematic and aspect information, as developed first by Kendrick, Maybeck, and Reid [121] and extended in [120, 4, 209, 63]. However, in general these methods require reasonably good *a priori* knowledge of the target class – if kinematic/aspect tracking is attempted with an incorrect assumption of target class, the tracking filter may quickly fail, due to irreconcilable differences between (1) what the filter expects to observe and (2) what it actually observes.

For our purpose of object recognition, however, we will *exploit* this tendency for tracking failure with improper object class associations. These failures will be identified using the classical state estimation techniques of filter *residual sequence* and state-reasonableness monitoring – the correct association (object class) is presumed to be the one that does not fail, or is the best among those that do not fail.

*1.2.5 Development Plan.* Chapter II discusses the state of the art in pattern recognition, target tracking and sensor fusion. Chapter III will apply Bayes' Rule to develop a motion fusion framework in which both DP sequence comparison-based and kinematic/aspect tracker-based *motion fusion* object recognition algorithms can be placed. These motion fusion methods will be shown to be forms of *syntactic* pattern recognition, as opposed to the decision theoretic or heuristic methods of pattern recognition [212] more commonly applied in tactical target recognition. Feature or signature observable measurements, kinematic measurements, and *a priori* knowledge of object dynamic limitations for known classes are used as independent sources of information, and the joint likelihood of those events is exploited in any of several ways to form *generalized likelihood functions*, the highest value of which will indicate the correct association of unknown object and known object class.

Chapters IV and V will develop aspects of the ideas in Chapter III in detail, and will show simulation results to support theory. Chapter VI will link ideas from Chapter III with developments in Chapters IV and V to show how these developments can be exploited in a number of directions.

The output of this fusion process over several models is the one model for which the information from all sources – kinematic, feature observable, and *a priori* model knowledge – is most consistent (a form of *maximum likelihood* multisensor fusion for object recognition). Chapter VI will discuss applications for this approach. For example, we will define paths by which to attack some problems in tactical target recognition now considered to be unsolvable.

We will also show that, since kinematic-feature observable fusion algorithms can be considered as *maximum likelihood estimators* or *generalized likelihood functions*, these algorithms are suitable for analysis using the classical state/parameter estimation tool of *generalized ambiguity functions* [154, 198]. This tool will allow us to contrast motion-feature fusion performance using different feature observables, feature comparison metrics, and object kinematic model assumptions against the performance of (1) truly optimal (if unrealizable in practice, using *truth* information not available to an actual sensor) maximum likelihood techniques and (2) more conventional techniques. Significantly, evaluations with generalized ambiguity functions will lead naturally to a form of Cramér-Rao lower bound [184, 155] for the performance of object recognition algorithms.

### 1.3 Conclusion

In Sect. 1.2.1, we proposed the existence of observed and expected measurement sequences, and asked two questions: (1) Is it possible to compare these two sequences such that their origin target classes can be seen to be identical, if so, despite expansions and contractions? (2) Will a *sequence comparison* process reduce the likelihood of incorrect identifications, in comparison with other approaches? This research will show that the answer to both questions is *yes*.

The contributions of this research are (1) to fold *dynamic time warping* and related sequence comparison techniques as *maximum likelihood* methods into the structure of multisensor fusion, (2) to extend previous efforts in target tracking so as to provide dynamic object and target recognition, (3) to provide a Bayesian structure for understanding fusion of kinematic and feature observable information, and (4) to introduce the use of *generalized ambiguity functions* as an analysis technique for gauging the effectiveness of these and other multisensor fusion methods. These developments provide new theoretical understanding for object recognition, extend and unify the results of previous researchers, and provide paths for attacking currently unsolvable problems.

This chapter has defined the inspiration and justification for "motion warping" and other approaches for fusion of "kinematic" and "nonkinematic" information. In turn, it has outlined the structure within which this dissertation will develop and demonstrate the proposed approaches. The following chapter will examine related concepts in pattern recognition, target tracking, multisensor fusion, state and parameter estimation, and speech processing.



## II. Background

### 2.1 Introduction

The purpose of this chapter is to outline the five key disciplines which have contributed to, or form departure points for, the ideas embodied in this dissertation. These key disciplines are: (1) pattern recognition and automatic object recognition, (2) object tracking and state estimation, (3) dynamic programming sequence comparison, (4) multi-sensor fusion, and (5) the theory of generalized ambiguity functions. The field of multi-sensor fusion has deep roots in pattern recognition and object tracking, and has seen some application of dynamic programming. This research has greatly strengthened the utility of dynamic programming in multisensor fusion through the concept of "motion warping", and uses generalized ambiguity functions to quantify this utility.

Much of the following discussion is oriented toward a specific class of dynamic physical objects – specifically, objects that may be identified as *targets* in military operations or other remote sensing scenarios. The use of the word "target" rather than "object" in radar sensing is a historical custom – however, the reader should keep in mind that the fundamental issue in this research is to explore and exploit for recognition the generally characteristic physical relationships between object *behavior* and object *appearance* – whatever the object, sensor, or scenario.

Chapter III will develop the theory for a class of object recognition algorithms with this objective, and Chapters IV and V will show results from representative algorithms.

### 2.2 Pattern Recognition / Automatic Object Recognition

This section provides a structure for relating the accomplished research to elements in the vast array of existing concepts for pattern recognition and non-cooperative target recognition (NCTR). These fields have grown explosively and in innumerable directions in the past ten years, creating a need for articles and texts that simply define taxonomy without technical detail [28, 55, 89, 97, 218], and for conferences which bring together experts in different yet related fields who have worked theretofore in isolation from one another [42, 43, 44, 124, 126, 231].

The first two of the following subsections consider in turn the two primary classical pattern recognition approaches – decision theoretic and syntactic. Finally, we discuss concepts for object recognition with high range resolution (HRR) radar that form a departure point and comparison baseline for the original research described here. Additional material on pattern recognition of a generally tutorial nature is found in App. B.

**2.2.1 Decision Theoretic Pattern Recognition.** Decision theoretic [90] or “statistical” [92] pattern recognition methods begin with a choice of observable mathematical quantities or *features* by which to describe objects of interest. A particular set or vector of features associated with an object or measurement defines a point in a *feature space*.

As discussed in the previous chapter, features or feature observables for physical objects are generally functions of aspect angle. For an ideally-chosen feature set, the values of features for different classes and discrete orientations (generally a given number of aspect angle “bins”) will *cluster* separately in the feature space. Cluster dispersion for any particular object class/aspect angle will occur due to atmosphere and sensor noise, minor object variations, etc. Well-chosen features provide low *ambiguity* in the specific sense of that term used in [154:97]– that a particular combination of measured feature values tends to identify a particular object and its orientation uniquely.

A particularly desirable form of decision theoretic classifier is one which yields  $p(\omega_i | \mathbf{z})$  for *each* class  $\omega_i$  of  $J$  possible object classes – that is, the probability  $p$  that an observed object is a member of class  $\omega_i$ , given that we have observed some sensor measurement (vector) value  $\mathbf{z}$ . This probability is commonly obtained through Bayes’ Rule [212, 197]:

$$p(\omega_i | \mathbf{z}) = \frac{p(\mathbf{z} | \omega_i)p(\omega_i)}{p(\mathbf{z})} = \frac{p(\mathbf{z} | \omega_i)p(\omega_i)}{\sum_{j=1}^J p(\mathbf{z} | \omega_j)p(\omega_j)} \quad (2.1)$$

in which  $p$  represents a probability or probability density function (if the latter exists),  $\mathbf{z}$  represents an observed or measurement value, and  $\omega_i$  is the  $i$ -th object class (ignoring aspect angle variations for the moment) from a total of  $J$  classes.

Bayes’ rule defines the Bayesian classifier – a theoretically optimal classifier under certain conditions [212:113]. Use of this and related probabilistic or “parametric” clas-

sifiers require that we establish the probability or probability density  $p(\mathbf{z} | \omega_i)$  for each combination of possible measurement  $\mathbf{z}$  and known object class  $\omega_i$ . This is the process of density estimation, a form of "training" generally conducted by experiment using techniques discussed in detail in [212:134–154]. Density estimation is generally a nontrivial process and presents the major obstacle in implementing parametric classifiers [28].

For some actual observed  $\mathbf{z}$  from an unclassified object, the derived *classical likelihood*  $p(\mathbf{z} | \omega_i)$  of that observation from each class  $\omega_i$  provides the *new information* in a parametric classification process. Often,  $p(\mathbf{z} | \omega_i)$  is assumed to be described adequately as Gaussian. The *a priori* probability  $p(\omega_i)$  is, in military scenarios, often taken from order-of-battle information which estimates the relative fractions of numbers of each object (target) class expected on the battlefield. Under the (usually weak) assumption that we have this data for all potential objects, the rightmost denominator in the previous equation – the sum of numerator terms for all object classes – allows us to define the *a posteriori* probability measure  $p(\omega_i | \mathbf{z})$  for a given observed  $\mathbf{z}$ . The  $i$  for which  $p(\omega_i | \mathbf{z})$  is greatest is taken to identify the object class.

In the absence of specific *a priori* information on the object class distributions, we may choose to assume them equal, i.e., that  $p(\omega_i)$  is the same for each  $\omega_i$ . These terms then cancel from the numerator and denominator of Eqn. (2.1) to produce the often-seen form:

$$p(\omega_i | \mathbf{z}) = \frac{p(\mathbf{z} | \omega_i)p(\omega_i)}{p(\mathbf{z})} = \frac{p(\mathbf{z} | \omega_i)}{\sum_{j=1}^J p(\mathbf{z} | \omega_j)} \quad (2.2)$$

also called a (conditional) maximum likelihood classifier. In this form, the *a posteriori* probability *per se* is often irrelevant, since the selected class  $\omega_i$  will simply be the one which maximizes  $p(\mathbf{z} | \omega_i)$  – the *classical likelihood function*.

A further modification [10, 33] to this technique is to allow for generation of a  $\mathbf{z}$  by some source other than a known object class, such as noise and/or the set of all unknown object classes. This results in the addition of an equivalent " $J + 1$ st" object class.

For decision theoretic recognizers which record feature space information for each object as a function of aspect angle (generally a finite number of aspect angle bins, as noted above), the "best" feature match for any given object model is often used to provide a "best" estimate of object aspect angle – a *pose estimate*. Note that the pose estimate *per se* is an angle value, and provides no information about the closeness of the model-object match in a *feature space distance* sense. A sequence of pose estimates, or *pose estimate history*, is the result of matching a sequence of feature observable measurements over time to a given model.

Most object recognition approaches are "independent look" algorithms – that is, they do not consider the location of previous pose estimates in making the most current estimate. But, for many physical objects (and tactical targets in particular), transitions between pose estimates imply object motion and/or random fluctuations of some kind in the feature observable measurements. It would seem, therefore, that allowable pose estimate transitions should be restricted in some way, so as to be consistent with other information about each object model, as observed or known *a priori*. This observation, made by the author and independently by others [20, 136, 164], but little developed to date, is a prime motivation behind this research.

**2.2.2 Syntactic Pattern Recognition.** It happens that decision theoretic classifiers are ill-posed to derive information from the change of object observations *over time*. In our research, we will exploit these *time domain* relationships – therefore, we now consider *syntactic* or *structural* classifiers, a class of algorithms well-suited for considering time domain relationships. Ultimately, all of the new techniques in this research will be shown to be syntactic classifiers. The following points are taken largely from [212], with a flavor of more recent developments from [90, 161], and examples or elaboration by the author.

Fundamentally, decision theoretic techniques reduce all information about an unclassified object into *quantities*, and assign class membership based on information about corresponding quantities for known object classes. Syntactic pattern recognition, on the other hand, looks for structure or *order* in feature observables, and assigns class membership based on similar structure for some known class. In general, the input to a syntactic

classifier is a sequence of "features" (often called *primitives* or *terminals* in syntactic systems) from some unclassified object, which is compared to corresponding sequences for known object classes. This branch of pattern recognition arose from efforts to define mathematical models of grammar for computer analysis of human language – hence the incorporation of many terms commonly associated with languages.

First, consider an example to illustrate the difference between decision theoretic and syntactic classifiers. Suppose we wish to design a word recognizer, using individual letters as features. A simple decision theoretic classifier might then identify the word *CAT* by its position in some 26-dimensional alphabetic-quantity feature space corresponding to the presence of one *A*, one *C*, and one *T*. Unfortunately, completely unable to distinguish order, this classifier would be unable to discriminate the word *CAT* from the abbreviations for *Tactical Air Command* or *Air Training Command* (lest this proposal seem too ridiculous, note that a *language recognizer*, working in this way on relatively long passages of text, might well be a workable yet simple proposition). A syntactic word classifier, however, would distinguish between the three "words" by considering their structure, or order of the letters.

The following terms are basic to discussion of syntactic pattern recognition:

- (1) An *alphabet* is a finite set of symbols.
- (2) A *sentence* (also *string* or *word*) over an alphabet is any string of finite length composed of symbols from the alphabet.
- (3) A *language* is any set (finite or infinite) of sentences over an alphabet.
- (4) Each language has a unique *grammar*, which describes the structure of its sentences, and can be defined as the fourtuple  $(V_N, V_T, P, S)$ , where

$V_N$  is a set of *nonterminals* (variables);

$V_T$  is a set of *terminals* (constants);

$P$  is a set of productions or rewriting rules;

$S$  is the start or root symbol (often corresponding to an idea to be expressed).

Grammars in turn are assigned into one of four type categories (types 0, 1, 2, or 3) according to the kinds of productions allowed in each. Productions are really the rules of the grammar – in a language sense, they tell us how one set of variables can be replaced by another set of variables and/or constants. The most general forms of grammar are “unrestricted” or “type 0” grammars, for which any set of variables can be replaced by another set of variables and/or constants, or none at all, without regard to *context* or previous use of the chosen variables and constants – factors which are critical for the other grammar types.

The objective of a syntactic pattern recognition or classification process is to identify the language which generated a given (unclassified) string or sentence of terminals or features. This is done by analyzing the string to determine which grammar most probably generated the string – a process called parsing. Parsing techniques are classified as either top-down or bottom-up. In a bottom-up technique, we start with the sentence itself and attempt to subdivide it into units which reveal its grammatical structure. In a top-down technique, we start with each language/grammar set and attempt to construct the observed sentence, ultimately determining which grammar will most readily do so.

As a simple example of a top-down technique, suppose that we have a sequence of spoken sounds in an unknown language which we believe, based on *a priori* information, to have the English meaning, “*the car is broken.*” We wish to determine the origin language. Using a top-down parsing procedure, we start with each language and work down through the grammatical structure (variables like *noun*, *verb*, etc.) to generate the appropriate sequences (one per candidate language) of terminals or constants – sound primitives. Finally, using sequence comparison techniques such as those to be discussed in Sect. 2.4.2, we can compare each generated sequence in turn with the observed sequence, assigning the observed sequence to the language for which we find the closest sequence-to-sequence match.

In addition to the sequence comparison techniques referred to above for grammar determination, another parsing technique is provided by the use of *finite automata* [212,

70, 161]. A *finite automaton*  $\mathcal{A}$  over an alphabet  $\Sigma$  is defined as:

$$\mathcal{A} = (K, \Sigma, \delta, q_0, F) \quad (2.3)$$

where  $K$  is a finite, nonempty set of states,  $\Sigma$  is a finite *input alphabet*,  $\delta$  is a *mapping* of  $K \times \Sigma$  into  $K$ ,  $q_0$  in  $K$  is the *initial state*, and  $F \subseteq K$  is the set of *final states*.

In practice, one uses an automaton as a sort of “*matched filter*” to identify the grammar of a given string of terminals (constants/features/primitives) [70:190–214]. An automaton is devised for each grammar, and the string is “read” terminal-by-terminal by each automaton. Each terminal element in general causes a state change – if the automaton rests in one of its allowable final states after reading the string, the string is said to have been *accepted* by that automaton. The set of all sequences accepted by an automaton indeed defines a language, and there is a one-to-one correspondence between grammars and automata. The use of automata for language recognition constitutes a bottom-up parsing technique.

Straightforward extensions to these ideas lead to the concept of stochastic grammars and automata, reflecting, for example, the real world variations and associated probability densities with which any given idea can be expressed in a given language using different combinations of words and speaker pronunciations. Much more could be said about syntactic pattern recognition – we have discussed here only the topics and techniques needed for later development.

Syntactic pattern recognition approaches for recognition of dynamic objects and tactical targets are much less common than methods using decision theoretic or “statistical” approaches. A particular set of references showing recent efforts to use syntactic methods for radar signal classification is [49, 194] – an older reference (about which more will be said in Sect. 2.5.7 and elsewhere) is [136].

A significant development in syntactic pattern recognition for dynamic object recognition was the 1978 paper by Therrien [211]. He applied the concepts of linear predictive signature estimation and sequential hypothesis testing [216] to two-class discrimination. Therrien’s approach was drawn from the class of linear estimation techniques which make

predictions of the next anticipated measurement, based on past measurements, and compare the actual observed measurement to this prediction to derive corrections for the current estimate, followed by a prediction for the subsequent measurement, and so on.

This difference between predicted and observed measurements has been termed the *innovation* [118] or *residual* [153:218]. Under nonrestrictive conditions, for the correct association of estimator and observed process, as we will see in Sect. 2.3.1.3, the *residual sequence* has interesting properties which make it an indicator of the likelihood that the observed measurements were generated by an object with parameters like those of the estimator model.

The drawback to this approach is that dynamic objects and tactical targets in physical space seldom have sensor signatures that are amenable *per se* to linear predictive estimation, as we will discuss in Sect. 3.3. A major direction in this research, however, has been to define approaches for using techniques similar to those of Therrien where possible, and to define extensions useful where linear or quasi-linear methods do not apply.

In addition to linear predictive methods, sequential hypothesis testing can be shown to be applicable to the multisensor fusion algorithms defined in this research – determining how long sequence comparison or “motion warping” needs to continue in any object recognition scenario to make a decision with desired low probability of incorrect classification.

A recent development in the area of syntactic pattern recognition with tactical applications is the effort by Seibert and Waxman [199, 220], who trained neural nets to recognize 3-D objects (aircraft) using sequences of images over changing aspect angles. The referenced sources list related efforts. Seibert and Waxman do not formally refer to their efforts as syntactic in nature, but their descriptions of the factors which motivated them in their chosen direction speak eloquently, if implicitly, of the existence and utility of “grammars” governing the behavior of real objects over time. Other recent developments in the area of syntactic pattern recognition are mentioned in the following subsection.

### *2.2.3 Example: Target Recognition by High Range Resolution (HRR) Radar.*

The preceding part of this section has been oriented toward theoretical methods. Since the accomplished research used simulated high range resolution (HRR) radar as the fea-



ture observable sensor, in this final part of our object recognition overview, we will focus on current approaches to HRR radar target recognition. We will see that both decision theoretic and syntactic approaches have been applied to this problem. The research accomplished for this dissertation significantly increases the theory and practical scope of syntactic methods for HRR radar and other target recognition problems.

Using high frequency radar pulses with specialized wideband waveforms, it is possible to isolate radar returns as a function of distance (range) along the sensor-object vector [203, 204, 221, 213, 78, 164]. The returns are summed (as complex vectors) into "range bins" which partition the distance along the range vector: unlike conventional radar, HRR radar range bins are small with respect to actual object extent as projected along the range vector, perhaps several hundred range bins corresponding to the actual object extent. The "range sweep," or returned waveform function (radar cross section or "RCS" plotted as a function of range) is recognized to be characteristic of each object class at each particular aspect angle, principally due to peaks in the waveform representing collections of major scatterers at given distances along the sensor-object vector (although not generally at the same location on the body of the object, i.e., possibly separated in crossrange).

Fig. 2.1 shows a collection of three measured HRR radar signatures for an unidentified aircraft – the length of the figure is on the order of the size of an aircraft, so that the bin widths are on the order of 0.05 to 0.50 meters (the actual size of the aircraft is not available). The rearmost part of the target aircraft lies to the left. Each of these signatures is actually the result of summing returns from several dozen individual pulses taken over a period of much less than one second. More will be said about these signatures in Chapter V.

HRR radar signature generation can be treated theoretically as the result of convolving a sinc-shaped (i.e., a function of the form  $(\sin x)/x$ ) radar pulse in range/time (i.e., a rectangle in frequency with chosen center frequency and bandwidth) with an array of scatterers represented as impulse (i.e., Dirac delta) functions in range/time. As we will see in Chapter V, many effects observed in simulations and tests are readily explained from this perspective, at least to first order.

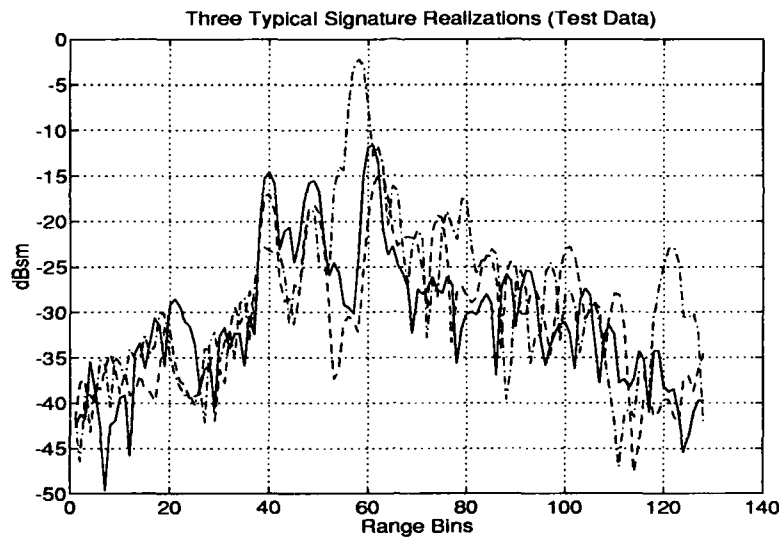


Figure 2.1. Three Typical High Range Resolution Radar Signatures – Rear Aspect (test data taken from tape GTw11Atran.dathr, results of [20], provided by [166], with further processing by the author)

HRR radar is subject to two primary error sources. First, the returns from scatterers in any given range bin have arbitrary phase, meaning that they will reinforce or cancel stochastically. Complete reinforcement in all range bins theoretically defines the maximum possible signal, or the “envelope” of the returned waveform. Total or near total cancellation in any given bin, on the other hand, produces a sharp downward spike called a “null.” The stochastic nature of this inter-scatterer phase problem is exacerbated for real objects, which vibrate as they move, causing scatterers to move relative to one another.

Second, signal paths that include more than one scatterer – i.e., paths which reflect from one scatterer to another (or more) and then back to the sensor – create returns that appear “later” or at longer range than we would expect from studies that do not consider multiple-scatterer returns. Since multiple-scatterer interaction models provide a combinatorial expansion in simulation complexity, many models do not address this problem [166, 64]. Cavities on the object are a particularly troublesome source of such noise.

It should also be noted that HRR radar does not provide "in-plane rotation-invariant" (see "PSRI" in App. A) signatures or features – that is, the HRR signature is a function of the "roll" angle of the target relative to the sensor around the sensor-to-target vector. This fact is due to the effects of waveform polarization at different target rotation angles. For small errors in sensor-target roll angle knowledge, effects on target recognition due to incorrect polarization assumptions are not considered separately from the other (more significant) error sources noted above.

HRR radar techniques are related to Inverse Synthetic Aperture Radar (ISAR) imaging techniques in the sense that ISAR uses the amplitude *and phase* of the return in each range bin, whereas classical HRR radar ignores phase information. Also, whereas HRR radar techniques yield an identification directly, classical ISAR yields only a crude image, which then is passed to an image-matching algorithm for actual identification [221, 52].

A contract between the USAF's Wright Laboratory and General Dynamics (GD) was one of the first detailed efforts in HRR radar, and GD's results as expressed in [20, 94, 78, 213] form the basis for the following part of this discussion. Related efforts are now underway with Hughes and Westinghouse under separate contracts [166].

The GD HRR recognition approach starts with the gathering of radar range sweeps for actual targets at discrete intervals ("granularity") of aspect angle and polarization. Ideally, we have multiple measurements for each aspect angle and polarization over the entire  $4\pi$  steradian extent of target aspect angle (i.e., the hypothetical target aspect angle sphere, as in Fig. 1.2). Each sweep is then processed to extract its significant peaks, recording them in terms of range (along the sensor-target vector) and amplitude. One set of such "reduced" measurements (one "reduced" range sweep per discrete aspect angle value) is chosen to represent the target library set ("set one"). The remaining reduced measurements, or members of "set two", are used in training the algorithm.

The "slide distance metric" [20, 94, 78] is used (1) in actual application or testing to define distances (i.e., measures of difference) between elements in set one (the library) and the measurements from an unclassified target, and (2) in training to define distances between elements in sets one and two. During the training process, comparisons of elements

in set one and set two are used to establish statistics whereby, in actual application, the slide distance results from a given measured signature can be used to define probabilities of class membership for the unknown target.

In application, the metric is repeatedly applied while "sliding" an unknown target's reduced sweep in range relative to a library reduced sweep to align the two range sweeps in each of four different ways – aligning the first peak of the unknown sweep to the first peak of the library sweep, aligning the second peak of the unknown sweep to the second peak of the library sweep, and so on through a "fourth peak-to-fourth peak" match. An acceptable peak match is defined as one for which the amplitude difference for the matched peaks is less than or equal to some defined value. Following the initial match, the remaining peaks in the two sweeps are paired, again subject to the amplitude difference criterion, and (since remaining peaks will not in general be aligned horizontally, i.e., in range) a horizontal difference criterion.

For a comparison between any two reduced sweeps, this process results in four candidate alignments, each of which involves matched peaks, unmatched peaks in the unknown sweep, and unmatched peaks in the library sweep. The slide distance  $d$  of each candidate alignment is computed using a formula given in [94]:

$$d = \frac{(T_1 + T_2 + T_3)^{1/2}}{W_{b-a}^{1/2}} \quad (2.4)$$

where:

$$T_1 = \frac{[\sum (W_{b-a} \times \delta_{bin}^2 \times \delta_{amp}^2)]}{N_{mpp}} \quad (\text{summation is over all matched peak pairs})$$

$$T_2 = F_{pen} \times [10 \log(R_u)]^2$$

$$T_3 = F_{pen} \times [10 \log(R_l)]^2$$

$$F_{pen} = \text{a constant penalty factor}$$

$$W_{b-a} = \text{bin-to-amplitude weight (a constant)}$$

$$\delta_{bin} = \text{the difference in range bin locations for two peaks that have been paired}$$

$$\delta_{amp} = \text{the difference in amplitudes for two peaks that have been paired}$$

$N_{mpp}$  = the number of matched peak pairs

$R_u$  = ratio of power of matched peaks in unknown sweep to power of all peaks in that sweep

$R_l$  = ratio of power of matched peaks in library sweep to power of all peaks in that sweep

and the "power" of a set of peaks, as discussed in the definitions for  $R_u$  and  $R_l$ , is defined as the sum of the peak amplitudes in square meters.

Finally, the smallest slide distance of the four candidate alignments is chosen as the final slide distance between the two reduced sweeps. Finer, bin-by-bin realignment and slide distance reduction is possible beyond this point, and is discussed in [20], but is not implemented in the code provided to this author. One may observe that this process achieves a kind of correlation between the two sweeps.

Another HRR radar signature metric is the MFML, or *Minimum Feature, Maximum Likelihood* metric [20] developed by Cyberdynamics under contract to General Dynamics. This metric has given good results in recent research by Cyberdynamics, subsequent to the original General Dynamics work [18]. Efforts to date by Westinghouse under contract to WL/AARA [166] have employed neural nets to analyze individual signatures (i.e., the classification "metric" is defined by the net during training).

It is significant for our purpose to note that most of the referenced sources [94, 78, 213] do not discuss target kinematics in any context, particularly with regard to determining the fundamental feasibility of the maneuvers which are implied by the sequences of aspect angles selected during the association process discussed in the preceding paragraphs. However, the final report for the General Dynamics effort [20] gives equations for "conical uncertainty," used to size search windows, and the report notes on page 4-2, with regard to the MFML metric, that "progression" (in aspect angle) of aspect angle (pose) estimates derived from the recognition process could be used to reduce the probability of misidentification, since the correct target model will show "smooth progression." The report notes that "this capability has not yet been implemented."

Finally, an additional metric is provided from references by Fukunaga [92], Mieras [164], Mitchell [166], and Stewart [207]. This metric is a *Mahalanobis* metric, as defined in [212], i.e., a quadratically-weighted squared distance of the form  $[\mathbf{z}_u - \mathbf{z}_\omega]^T \mathbf{S}_\omega^{-1} [\mathbf{z}_u - \mathbf{z}_\omega]$ , where  $\mathbf{z}_u$  is a vector of range bin returns for an unknown target,  $\mathbf{z}_\omega$  is a vector of mean range bin returns for a known aspect angle on a known target of class  $\omega$ , and  $\mathbf{S}_\omega^{-1}$  is the inverse covariance matrix corresponding to the mean vector  $\mathbf{z}_\omega$ . For a process  $\omega$  with the Gaussian classical likelihood  $p(\mathbf{z} | \omega)$  of mean  $\mathbf{z}_\omega$  and covariance  $\mathbf{S}_\omega$ , the Mahalanobis distance  $M$  is the key argument in the "log likelihood" of observing an event  $\mathbf{z}_u$ , or  $p(\mathbf{z}_u | \omega)$ . That is,  $\ln[p(\mathbf{z}_u | \omega)] = C - (0.5 \times M)$ , where  $C$  is a function of  $\mathbf{S}_\omega$  which can often be taken as a constant. The straightforward multivariate Gaussian origin of the Mahalanobis metric makes it readily applicable in probabilistic approaches, and this metric is the primary signature metric investigated in this research.

Despite their common use of this metric, the users noted in the previous paragraph have applied it in significantly different fashions. Fukunaga uses a high range resolution radar database gathered for two automobile classes (a sedan and a van) at various aspect angles to illustrate concepts in statistical pattern recognition. This work makes no use of motion information for *a priori* limiting (conditioning) of aspect angle. Unlike the other HRR radar efforts discussed prior to this point, the method of Mieras (Raytheon) [164, 165] for HRR radar aircraft target recognition is syntactic in nature, and intimately related to efforts in this dissertation (to be discussed further in Sect. 3.8.1). Maximum likelihood methods are proposed by Mieras to work around uncertainties in multiplicative scale factor and range registration (i.e., bin alignment on the unknown target). Finally, Stewart [207] reminds us that, where the covariance matrix is assumed to be diagonal with equal variance in each bin, the range registration process is a correlation in the range domain, and is thus more easily implemented in the frequency domain as a complex conjugate multiplication.

A fundamentally syntactic method for HRR radar target recognition using neural nets was proposed by Farhat [81]. He defined neural network classifiers using inputs of "sinograms," or sequences of high range resolution radar sweeps, as generated by *constant angular rate* changes of target aspect angle with respect to the sensor (i.e., as with the motion of a target model on a turntable). Farhat's published work does not discuss,

however, how he would deal with all of the possible "sinograms" that could be generated by all possible targets moving over all possible (not generally constant rate) trajectories – nor does he address the use of kinematic information available from the radar sensor. Presumably, a properly structured and trained neural net would *generalize* to some degree, but Farhat's approach does not clearly offer strong promise of being able to assign an observed sequence of range sweeps *unambiguously* to the correct target class.

Air Force Institute of Technology students DeWitt [68] and Kouba [128] have recently investigated the use of syntactic techniques (hidden Markov models and recurrent neural nets, respectively) for HRR radar target recognition. It is believed that their methods could be used in combination with concepts discussed in this research to design fully functional automatic object recognition systems.

*2.2.4 Pattern Recognition – Conclusion.* This completes our overview of pattern recognition and conventional approaches to automatic object recognition. Following sections will consider theory and techniques that we will ultimately fold back into object recognition algorithms, in attempts to make better use of information that is often available, but ignored in decision making.

### *2.3 Object Tracking / State Estimation*

The science of object tracking and state estimation grew out of efforts to use discrete radar observations to develop continuous object tracks, or estimates of object kinematic state. The research described in this dissertation in a sense *reverses* those efforts, using the kinematic state estimates to recognize the object class that generated the observations. First, it is essential to understand what information is available from object tracking.

The most significant milestone in object tracking development was the development of the Kalman filter [113, 119, 153, 198] for optimal state estimation of systems that can be adequately described by linear dynamics models with forcing functions of (heuristically) white Gaussian noise, given measurements corrupted by the same form of noise. The linear system structure of the Kalman filter made it immediately applicable to multisensor fusion of kinematic information, and special forms – the linearized and extended Kalman filters –

were developed to handle nonlinear dynamics and/or measurement equations [154]. With particular choices of models and noise assumptions for given objects, the extended Kalman filter forms the core of nearly every object tracking software system in use today.

**2.3.1 The Kalman Filter and Other Estimators.** The purpose of this section is to delineate the equations for the conventional Kalman filter and the modifications to those equations necessary in the case of the extended Kalman filter. The form of the equations will be that for a continuous dynamic system for which discrete (sampled data) measurements are available (the most common case in general and for tracking systems in particular). The last portions of this section discuss optimal smoothing of Kalman filter estimates, parameter estimation with multiple model Kalman filters, and nonlinear estimators. This information is taken from texts by Maybeck [153, 154], in some cases as previously summarized by the author [143].

**2.3.1.1 The Kalman Filter.** The Kalman filter derives the optimal estimate for a linear dynamic system described by:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \quad (2.5)$$

and which has discrete linear measurements corrupted by white Gaussian noise:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \quad (2.6)$$

where:

$\mathbf{x}$  = an  $n$ -dimensional system state vector

$\mathbf{F}$  = an  $n \times n$ -dimensional plant dynamics matrix

$\mathbf{B}$  = an  $n \times r$ -dimensional deterministic input matrix

$\mathbf{u}$  = an  $r$ -dimensional vector of deterministic inputs

$\mathbf{G}$  = an  $n \times s$ -dimensional dynamics driving noise distribution matrix

$\mathbf{w}$  = an  $s$ -dimensional vector of white Gaussian dynamics driving noises



$\mathbf{z}$  = an  $m$ -dimensional measurement vector

$\mathbf{H}$  = an  $m \times n$ -dimensional measurement matrix

$\mathbf{v}$  = an  $m$ -dimensional white Gaussian noise vector

$t$  = time (function argument)

$t_i$  =  $i$ th sample time (function argument)

The white Gaussian noise vectors  $\mathbf{w}(t)$  and  $\mathbf{v}(t_i)$  are assumed to be zero-mean, independent, and to have covariances defined by:

$$E[\mathbf{w}(t)\mathbf{w}^T(t+\tau)] = \mathbf{Q}(t)\delta(\tau) \quad (2.7)$$

and

$$E[\mathbf{v}(t_i)\mathbf{v}^T(t_j)] = \mathbf{R}(t_i)\delta_{ij} \quad (2.8)$$

where:

$E$  = the expectation operator

$\mathbf{Q}(t)$  = an  $s \times s$ -dimensional positive semidefinite matrix describing the strength of the dynamic driving noise vector,  $\mathbf{w}(t)$ , at time  $t$

$\mathbf{R}(t_i)$  = an  $m \times m$ -dimensional symmetric positive definite matrix – the covariance of the measurement error at time  $t_i$ . See [153:216] for further discussion on the assumption of positive definiteness for  $\mathbf{R}(t_i)$

$\delta(\tau)$  = the Dirac delta function, equal to infinity for  $\tau = 0$  and zero elsewhere, with area of unity when integrated over all  $\tau$  (units of  $time^{-1}$ )

$\delta_{ij}$  = the Kronecker delta, equal to one for  $i = j$ , zero otherwise (dimensionless)

For these systems, the optimal estimate is computed after each measurement as:

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i) [\mathbf{z}_i - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)] \quad (2.9)$$

and the estimated covariance of the (zero-mean) error for this estimate is:

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\mathbf{P}(t_i^-) \quad (2.10)$$

where:

$\hat{\mathbf{x}}(t_i^+)$  = estimated value of system state variables of interest after measurement update at time  $t_i$

$\hat{\mathbf{x}}(t_i^-)$  = estimated value of system state variables before update at time  $t_i$

$\mathbf{K}(t_i)$  = Kalman filter gain at time  $t_i$

$\mathbf{z}_i$  = observed measurement values at time  $t_i$

$\mathbf{H}(t_i)$  = measurement matrix at time  $t_i$  (post-multiplied by  $\hat{\mathbf{x}}(t_i^-)$  to give the predicted measurement value for time  $t_i$ )

$[\mathbf{z}_i - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)]$  = the filter residual, or difference between observed and predicted measurements at time  $t_i$

$\mathbf{P}(t_i^+)$  = estimated error covariance after measurement update at time  $t_i$

$\mathbf{P}(t_i^-)$  = estimated error covariance before measurement update at time  $t_i$

and  $\mathbf{K}(t_i)$  is found using the relation:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T(t_i) [\mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}^T(t_i) + \mathbf{R}(t_i)]^{-1} \quad (2.11)$$

for which all elements are defined above.

The elements of  $\hat{\mathbf{x}}$  and  $\mathbf{P}$  at time  $t_i^-$  are obtained by propagating the estimates of those quantities forward in time from the previous set of values according to the state dynamics:

$$\hat{\mathbf{x}}(t | t_{i-1}) = \mathbf{F}(t)\hat{\mathbf{x}}(t | t_{i-1}) + \mathbf{B}(t)\mathbf{u}(t) \quad (2.12)$$

and

$$\dot{\mathbf{P}}(t | t_{i-1}) = \mathbf{F}(t)\mathbf{P}(t | t_{i-1}) + \mathbf{P}(t | t_{i-1})\mathbf{F}^T(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) \quad (2.13)$$

where  $\hat{\mathbf{x}}(t | t_{i-1})$  = the estimated values of system state variables at time  $t$  based on measurements through time  $t_{i-1}$ , and the meanings of other variables may be inferred from earlier discussion. The initial state value is modelled as a random vector normally distributed with mean  $\hat{\mathbf{x}}_0$  and covariance  $\mathbf{P}_0$ . Often in the case of radar tracking,  $\hat{\mathbf{x}}_0$  is derived from the initial measurements and  $\mathbf{P}_0$  is a function of the error statistics of those measurements [10:80-82] [35:153-154] [143].

Note in particular that the accuracy of a Kalman filter state estimate so generated is completely dependent on the extent to which the filter dynamics, measurement, and noise statistical models reflect the true processes. In most cases, the true processes are either not fully understood or cannot be modelled exactly, or to do so would result in unacceptable filter computational requirements. In either case, the performance of a purposefully reduced-order and simplified filter can only be proven by actual use, but ideally it should be tested prior to implementation by simulating operation with a high fidelity truth model [153:289-291].

Consider, on the other hand, a process for which dynamics and/or measurements cannot be described by the linear Eqns. (2.5) and (2.6) above. An "extended" Kalman filter routine can be written for a system in which it is possible to express dynamics by an equation of the following more general form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \quad (2.14)$$

where:

$\mathbf{x}$  = an  $n$ -dimensional system state vector

$\mathbf{f}$  = an  $n$ -dimensional vector dynamics function

$\mathbf{u}$  = an  $r$ -dimensional vector of deterministic inputs

$t$  = time

$\mathbf{G}$  = an  $n \times s$ -dimensional dynamics driving noise distribution matrix

$\mathbf{w}$  = an  $s$ -dimensional vector of white Gaussian dynamics driving noises

and where measurements can be expressed as:

$$\mathbf{z}(t_i) = \mathbf{h}[\mathbf{x}(t_i), t_i] + \mathbf{v}(t_i) \quad (2.15)$$

where  $\mathbf{h}$  = an  $m$ -dimensional vector measurement function, and other values are as defined earlier. Note that for this development, the zero-mean white Gaussian noises  $\mathbf{w}(t)$  and  $\mathbf{v}(t_i)$  are, as before, required to be independent of each other and to enter the system linearly. As is the case with the conventional (linear) Kalman filter, the extended Kalman filter for the case of continuous dynamics and discrete updates is implemented as a series of alternating updates and propagations of the state estimate and filter-computed error covariance.

The updated state vector and covariance matrix are propagated forward in time to the next update using the relations:

$$\hat{\mathbf{x}}(t | t_{i-1}) = \mathbf{f}[\hat{\mathbf{x}}(t | t_{i-1}), \mathbf{u}(t), t] \quad (2.16)$$

and

$$\begin{aligned} \dot{\mathbf{P}}(t | t_{i-1}) = \\ \mathbf{F}[\hat{\mathbf{x}}(t | t_{i-1}), t] \mathbf{P}(t | t_{i-1}) + \mathbf{P}(t | t_{i-1}) \mathbf{F}^T[\hat{\mathbf{x}}(t | t_{i-1}), t] + \mathbf{G}(t) \mathbf{Q}(t) \mathbf{G}^T(t) \end{aligned} \quad (2.17)$$

where all quantities have been defined earlier except:

$$\mathbf{F}[\hat{\mathbf{x}}(t | t_{i-1}), t] = \text{an } n \times n \text{ matrix of partial derivatives, } \left. \frac{\partial \mathbf{f}[\mathbf{x}, \mathbf{u}(t), t]}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(t | t_{i-1})}$$

Updates are performed using the following equations:

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i) \{ \mathbf{z}_i - \mathbf{h}[\hat{\mathbf{x}}(t_i^-), t_i] \} \quad (2.18)$$

and

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i) \mathbf{H}[\hat{\mathbf{x}}(t_i^-), t_i] \mathbf{P}(t_i^-) \quad (2.19)$$

where:

$$\mathbf{h}[\hat{\mathbf{x}}(t_i^-), t_i] = \text{the predicted measurement value at time } t_i$$

$\{z_i - h[\hat{x}(t_i^-), t_i]\}$  = the filter residual, or difference between observed and predicted measurements at time  $t_i$ , in the form appropriate for the extended Kalman filter

$H[\hat{x}(t_i^-), t_i]$  = an  $m \times n$  matrix of partial derivatives,  $\frac{\partial h[x, t_i]}{\partial x} \big|_{x=\hat{x}(t_i^-)}$

and  $K$  is computed as given earlier using the value of  $H$  computed above. Other values are as defined earlier.

The matrix partials  $F$  and  $H$  represent the successive relinearization about newly declared nominal values, i.e., the latest best estimates of system state. Note therefore that, unlike conventional (and linearized) Kalman filters, the extended Kalman filter relations for updating and computing the filter-computed error covariance are functions of the estimate values. This dependence is unfortunate in that it forbids calculation of the actual filter gains and covariances *a priori*, an on-line computation-saving process which can be done for the linear filter. Pre-computed and constant-gain extended Kalman filters have been constructed using gain calculations from reasonably well-known nominal state trajectories, however [154:57].

**2.3.1.2 Forward-Backward Kinematic State Estimation: Smoothing.** In most tracking applications, we can accept a current estimate of the system (object) state, based or conditioned on the prior and current measurement history. In some applications, however, we need to define the best estimate of system state at each point in time along some portion of a trajectory, with the estimate at any time conditioned not only on prior and current measurements, but also on measurements taken *after* that time. The process of obtaining this estimate is called *smoothing* [154, 159, 160]. A typical requirement for smoothing arises in post-test analysis for antiaircraft missile test intercepts (without war-head detonation), where a best estimate of miss distance is required for lethality assessment, based on the complete measurement history of the engagement, i.e., for measurements both before *and after* the missile passes by the target aircraft.

An analogous motivation will exist for the research discussed here – in part, this research requires an estimate of aspect angle based solely on aircraft kinematics (more precisely, measurements of translation states). Under normal circumstances, as will be discussed in Chapter V, aircraft orientation can be reasonably well estimated as a function

of aircraft velocity and acceleration – acceleration commands (implemented directly by change of pitch, roll, yaw, thrust, or, if the design allows, drag) being an unknown input by the pilot. Our kinematic measurements, however, are generally of position only (perhaps position and doppler-derived velocity), and estimating velocity and acceleration inherently require us (at least conceptually, if not explicitly in the algorithm itself) to differentiate our position information once and twice, respectively. This implicit twofold differentiation and unpredictable pilot inputs mean that our ability to estimate acceleration at some time  $t_i$  from prior and current position measurements is quite limited. If we are willing to wait, however, measurements of position for times *beyond* time  $t_i$  will reveal much about the acceleration and orientation *at* time  $t_i$ . Thus, we may choose to make a series of radar measurements for an aircraft undergoing some maneuver, and then estimate the most likely aspect angle for each point in time during that maneuver, conditioned both on measurements before and shortly after that time point. For this purpose, we will require a *smoother*.

There are three main forms of smoothers, based on measurement information available and state estimate desired [154:1–18]: fixed interval, fixed point, and fixed lag. For the fixed interval smoother, we assume the availability of a set of measurements over some time interval, and determine the best estimate of the state for each point in time during that interval, conditioned on all of the measurements taken during that interval. For a fixed point smoother, we seek an estimate of the state at one time  $t_p$  only, conditioned on measurements before and after time  $t_p$  up until the most recent time, continuing to improve the estimate with measurements until some final time  $t_f$ . Finally, for a fixed lag smoother, we find an estimate for the state at each point along the trajectory, conditioned on measurements prior to that point in time and measurements for some fixed time interval (lag) beyond.

For the purpose considered here, the last form, or fixed lag smoother, is most relevant, and only this form will be discussed here. We will generally expect to maintain track on an aircraft target for a short time before selecting a time period over which to perform a target recognition process (a period selected, for example, because the target executes a *revealing* maneuver – one for which we expect significant differences between signatures

for potential target classes). Information collected prior to the period of interest should be made available to the estimator. The reader might point out that this implementation by definition imposes an added time lag beyond that required for a fixed interval smoother, but if this time lag is critical, we have options for producing smoothed estimates of the state for times during the lag period as well (e.g., a set of fixed point smoothers for constantly changing points in time during the lag).

Equations and procedures for implementing the fixed interval smoother are shown in App. C in the interest of saving space here. These equations are provided in linear Kalman filter format, as presented in [154:16–17], but no significant difficulties are encountered in applying them under extended Kalman filter assumptions.

*2.3.1.3 Residual Analysis and Multiple Model Filtering.* A well-known application for Kalman filters [154:129–136] is the estimation of parameters for state dynamics (“plant”), measurement, and/or noise models, simultaneously with state estimation itself, for systems from which measurements are available, but for which true parameter values are unknown. Note that the distinction between states and parameters made here is taken from Maybeck [154:69], essentially that parameters and states taken together are variables that define a system and its activity, but parameters, if they vary at all, vary slowly enough to be considered essentially constant for periods of interest to the estimator. In our case, for example, we might recognize the physical structure of a tactical target as defining its parameters, while its six degree-of-freedom (“6 DOF” – three translational and three angular degrees of freedom) motion through physical space over time defines its state history.

When a discrete, finite number of likely parameter vectors (values) can be identified *a priori*, we can define a set of Kalman filters according to these model parameters, and use the filters in a multiple model adaptive estimator (MMAE) structure to estimate the true parameter vector for the unknown system. We will use the variable  $\omega$  here to represent sets or vectors of parameter values, recognizing that  $\omega$  was used earlier in this chapter to represent object classes for pattern recognition. This choice is *completely intentional*. One of the basic themes of this research is that an “object” is, in an abstract sense, simply a

point or vector of parameters in some infinite dimensional parameter space, which we must approximate or model as finite dimensional in practice.

Parameter identification using sets of Kalman filters, or multiple model estimation, is accomplished by analyzing the residual sequences from the respective filters. Filter residuals are the quantities  $\{z_k - H(t_k)\hat{x}(t_k^-)\}$ , for a linear Kalman filter, or  $\{z_k - h[\hat{x}(t_k^-), t_k]\}$  for an extended Kalman filter – that is, the differences between actual and expected measurements at each time  $t_k$ . These residuals were respectively the quantities in square brackets in Eqn. (2.9) and braces in Eqn. (2.18) which provide new information to recursively update the current state estimate.

For a linear filter with the classical Gaussian noise assumptions and proper model parameters  $\omega_i$ , the residual sequence over time can be shown to be zero mean, white (independent in time), and Gaussian, with covariance  $A_i(t_k) = H(t_k)P(t_k^-)H^T(t_k) + R(t_k)$ . Equivalently, we can say that we expect the measurements  $z_k$  to have the probability density function or likelihood  $p(z_k | Z_{k-1}, \omega_i)$ , which has a mean of  $H(t_k)\hat{x}(t_k^-)$  and covariance  $H(t_k)P(t_k^-)H^T(t_k) + R(t_k)$ . These properties tend to hold true (to first order) as well for cases well-estimated by linearized and extended Kalman filters (in the latter case, the mean is given by  $h[\hat{x}(t_k^-)]$  rather than  $H(t_k)\hat{x}(t_k^-)$ ).

It is extremely important to recognize that the likelihood  $p(z_k | Z_{k-1}, \omega_i)$  is explicitly a measure of the *joint likelihood* that the system (parameter set  $\omega_i$ ) modelled by the Kalman filter could simultaneously output the elements of the observed measurement vector  $z_k$ , having previously output the measurement vector sequence  $Z_{k-1}$ . The utility of residual analysis is that measurements from a system with parameters  $\omega_j$ , processed by a filter designed for parameters  $\omega_i \neq \omega_j$ , will not in general exhibit the proper residual mean (i.e., zero) or covariance. This deviation betrays an improper model choice.

At this point, we need to observe that this approach is intimately related to the method of Therrien [211] [9:177–178] noted in Sect. 2.2.2 above. Therrien observed that a wider class of tools than simply the Kalman filter are available which produce residual sequences having the properties noted above. His approach, however, was oriented toward linear prediction of sensor signatures for residual analysis. In cases of interest to us, as we



will see in the following chapter, linear prediction of sensor signatures is not generally feasible, yet we desire to exploit the information contained in these signatures. In subsequent chapters, this research will extend residual analysis for object recognition using kinematics and sensor signatures, where linear prediction of signatures is not reasonable or feasible.

Continuing the discussion of classical multiple model parameter estimation, note that given *a priori* knowledge of the likelihood of parameter sets  $\omega_j$ , Bayes' Rule can be used to find an *a posteriori* probability of the presence of each parameter set, conditioned on the observed measurements and *a priori* knowledge:

$$p(\omega_i | \mathbf{Z}_k) = \frac{\{\prod_{n=1}^k [p(\mathbf{z}_n | \mathbf{Z}_{n-1}, \omega_i)]\} p(\omega_i)}{\sum_{\omega_j, j=1}^J \{\prod_{n=1}^k [p(\mathbf{z}_n | \mathbf{Z}_{n-1}, \omega_j)]\} p(\omega_j)} \quad (2.20)$$

for  $i = 1, 2, \dots, J$ , where:

$\mathbf{Z}_{n-1}$  = a set of  $n - 1$  measurement vectors  $\mathbf{z}$

$\mathbf{z}_n$  = a vector of measurements available at time  $t_n$

and other variables are as defined earlier.

An iterative version of Eqn. (2.20) which converges in practice [154:133] to an *a posteriori* probability of one for the true parameter set  $\omega$  is given by:

$$p(\omega_i | \mathbf{Z}_k) = \frac{p(\mathbf{z}_k | \mathbf{Z}_{k-1}, \omega_i) p(\omega_i | \mathbf{Z}_{k-1})}{\sum_{\omega_j, j=1}^J p(\mathbf{z}_k | \mathbf{Z}_{k-1}, \omega_j) p(\omega_j | \mathbf{Z}_{k-1})} \quad (2.21)$$

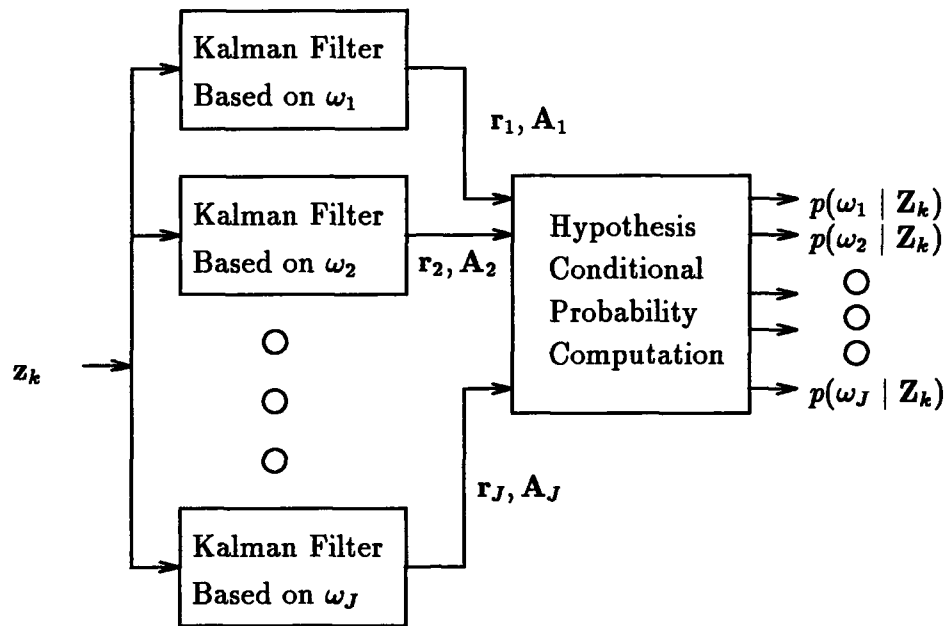
where:

$p(\omega_j | \mathbf{Z}_0) = p(\omega_j)$  (the *a priori* probability of class  $\omega_j$ )

and other variables are as defined earlier.

Note that in the absence of specific information on  $p(\omega_j)$  for each  $j$ , they may be assumed equal, or effectively ignored. This development is entirely analogous to that of Eqn. (2.2).

Conceptually, Fig. 2.2 expresses the way in which this multiple model parameter estimation process is conducted. Note that this figure and the preceding discussion have



Where:  $z_k$  = the  $k$ -th element in measurement sequence  $Z_k$   
 $\omega_j$  = the  $j$ -th class of  $J$  target classes  
 $r_j$  = the residual vector for class  $j$   
 $A_j$  = the residual covariance for class  $j$   
 $p(\omega_j | Z_k)$  = the *a posteriori* probability of class  $j$

Figure 2.2. Multiple Model Estimation Algorithm

not considered the fact that it is possible and often desirable to obtain a single "overall" estimate for the state  $\hat{x}_k$  and parameter set  $\omega$ , conditioned on the individual state estimates provided by all of the individual estimators, and the likelihoods of their respective parameter sets. This capability will not be required for our purposes, but is covered in [154:129-136].

An extension to the basic concept of parameter estimation through residual analysis and multiple model filtering is that of "state reasonableness checking" [59, 157]. If the filter designer knows that a particular state in a filter designed for a particular parameter set should have a certain nominal value, he may heuristically treat the difference between

the estimated value and the nominal value like a residual, with variance given by the appropriate element in the filter covariance matrix  $\mathbf{P}(t_k^+)$ . Modifying Eqn. (2.20), this results in a Bayesian parameter estimation format of the form:

$$p(\omega_i | \mathbf{Z}_k) = \frac{\{\prod_{n=1}^k [p(\mathbf{z}_n, \hat{\mathbf{x}}_n^r | \mathbf{Z}_{n-1}, \omega_i)]\} p(\omega_i)}{\sum_{\omega_j, j=1}^J \{\prod_{n=1}^k [p(\mathbf{z}_n, \hat{\mathbf{x}}_n^r | \mathbf{Z}_{n-1}, \omega_j)]\} p(\omega_j)} \quad (2.22)$$

where:

$\hat{\mathbf{x}}_n^r$  = state variable estimates at time  $t_n$  for which we wish to check reasonableness (generally a *proper subset* of the filter state vector estimate  $\hat{\mathbf{x}}(t_n)$  – hence the superscript  $r$ ).

and other variables are as defined earlier. A convergent form of this parameter estimator can be defined analogously to Eqn. (2.21).

A further extension to this basic multiple model estimator approach provides for the parameters of the estimators themselves to change or adapt based on observed *a posteriori* parameter class probabilities [154:136] [147]. This is the most general form of the multiple model approach, or the true multiple model *adaptive* estimator.

**2.3.1.4 Nonlinear Filters and Further Developments.** We have seen that the extended Kalman filter provides a means to apply linear filter theory to systems with nonlinear state dynamics and/or measurements of the form given in Eqns. (2.14) and (2.15), by successively relinearizing about the latest best estimate. The prices paid for this facility are several: the resulting estimate is neither “optimal” with respect to every reasonable definition of optimality [153:205] (e.g., the extended Kalman filter generally provides a biased estimate of the true state [154:52]), nor is filter stability guaranteed (as is linear filter stability under nonrestrictive assumptions [154:24]), nor are filter gains precomputable.

Consider, on the other hand, an arbitrary nonlinear system of the form specified by the following heuristic (since its form is incorrect for use in a proper stochastic integral) nonlinear stochastic differential equation (where all variables have been previously defined) [154:159–202]:

$$\dot{\mathbf{x}} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t] \quad (2.23)$$

It can be shown that solutions to this state equation will not in general be *Markov* (specifically we refer here to a Markov-1 system): i.e., knowing the value of  $\mathbf{x}$  at some time  $t_{i-1}$  does not provide as much information about  $\mathbf{x}$  at some later time  $t_i$  as we might obtain with knowledge of  $\mathbf{x}$  at  $t_{i-1}$  and at earlier times. This failure to be Markov means that we cannot propagate a conditional density of the state estimate, such as we did in effect for the linear Gaussian case using Eqns. (2.12) and (2.13). Thus, although we can use Bayes' Rule to define the change in our estimate from a measurement at any given time, we cannot properly propagate that information forward to the next measurement time.

However, for a system which can be described by an *Itô* stochastic differential equation of the (rigorous) form:

$$d\mathbf{x}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t]dt + \mathbf{G}[\mathbf{x}(t), \mathbf{u}(t), t]d\beta(t) \quad (2.24)$$

or, heuristically,

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{G}[\mathbf{x}(t), \mathbf{u}(t), t]\mathbf{w}(t) \quad (2.25)$$

(where  $d\beta(t)$  is the stochastic differential of a vector Brownian motion process and all other variables have been previously defined), then the solution  $\mathbf{x}(t)$  is Markov (although not generally Gaussian), and it is conceptually possible to propagate the conditional density, conditioned on the previously known state value, using the *forward Kolmogorov equation*, or *Fokker-Planck equation* [154:192-215]. Given measurements of the form in Eqn. (2.15), Bayes' rule, and the *Chapman-Kolmogorov equation*, we can theoretically define an optimal nonlinear estimator of the conditional density of states, conditioned on the observed measurement time history [154:212-215]. However, this optimal estimator generally will be infinite dimensional, and therefore impossible to implement in practice.

However, if we choose to approximate conditional densities of  $\mathbf{x}$  with a finite number of moments or an assumed Gaussian form (noting that Gaussian densities require but two

moments for a complete density function characterization), we can define *conditional moment estimators* [154:215–238], higher order filters that can be implemented in a practical fashion (at least through fourth moments) to provide improved performance over the extended Kalman filter. Under certain assumptions and/or conditions (ignoring higher order derivatives, etc.) [154:223], since Eqns. (2.5) and (2.14) are Itô stochastic equations with the general form of Eqn. (2.25), these higher order filters reduce to the extended Kalman filter or even linear filters.

A particularly efficient modification to the extended Kalman filter, providing generally intermediate performance between the basic extended Kalman filter and the nonlinear filters discussed above, is the addition to the extended Kalman filter of so-called “bias correction terms” [154:215–238]. These terms are in fact key terms from the higher order filter expressions. In every case noted so far, however, improved performance is gained only at the expense of additional computation. Readings by the author appear to show in general that, thus far in practice, systems designers find the basic extended Kalman filter adequate for most purposes, and do not incur the expense of going to higher order nonlinear filters.

A number of other routes to nonlinear filters are available, as discussed in [154:239–259]. In particular, recent work by Bishop [32] demonstrated application of a *geometric nonlinear filter* to aircraft tracking. His effort is of special interest because this author’s extensive literature survey (see previous references in this section) indicates that Bishop’s work is perhaps the only fundamentally new approach to *kinematic measurement-only* aircraft tracking for fire control (gun aiming) in the last ten years. This effort is discussed in more detail in App. C.

**2.3.2 Object Tracking with Kinematic Measurements Only.** Prior to the landmark effort by Kendrick *et al.* [120, 121], object tracking filters or target state estimators were designed to use only “kinematic” measurements – defined here as in Chapter I to be measurements of the translation of the object centroid through physical (generally three-dimensional) space. This information took the form of measurements of object range, range rate, pointing angle, and angle rate, as provided to varied extents by radar, passive optical

or infrared tracking systems, or laser. An excellent historical discussion of the material presented here is found in the 1984 work by Chang and Tabaczynski [50].

*2.3.2.1 Common Filter Models for Tactical Targets.* Given that an extended Kalman filter has been selected for a particular tracking problem, the designer is faced with three decisions or choices – reference frame in which to implement the filter, object (behavior) model, and measurement noise assumptions. We will address the more common choices for each in turn, again drawing heavily from a previous work by the author [143] – a recent (1990) work giving more detail is [35].

The two principal frames considered for airborne tracking are the line-of-sight (LOS) or antenna frame, and the inertial or inertial measurement unit (IMU) frame. The LOS frame must be further classified as constantly rotating with respect to inertial space or space-stabilized but impulsively realigned between measurements [229]. For radar tracking, the LOS frame is easily related to the “measurement coordinate” frame [35:183] in which a tracking problem may be directly posed in the  $m$ -dimensional measurement vector space. Variations in IMU frame mechanizations depend primarily on the Cartesian frame with respect to which the designer desires to define accelerations. Common choices of inertial reference frame include the locally-fixed North-East-Down *site* or *navigation* frame, as for ground or local air vehicle tracking, or the earth-centered *geocentric* frame (fixed with respect to the stars), as for satellite tracking.

Since object dynamics are more easily expressed in an inertial frame, the use of an inertial frame often makes the object state equation (Eqn. (2.14)) more tractable, and, with certain dynamics models may even make it linear, as in Eqn. (2.5). With radar measurements of range and (small) error angles in azimuth and elevation, and perhaps range and/or angle rates, the LOS frame, on the other hand, can provide a linear measurement equation (errors induced by LOS frame implementations, and their compensation, are discussed in [35:184–188]). Linear relations are desirable in extended Kalman filter formulation where possible because they reduce the computationally expensive need to compute partial derivatives. In usual practice, the computationally more attractive filter design is often one with linear state dynamics and nonlinear measurements. This arises because

the dynamics relations are generally computed much more often in an extended Kalman filter than the measurement relations – to maintain high accuracy in numerical integration, dynamics are propagated sequentially over many subintervals of the measurement sample period. This is notably often the case for radar tracking.

The choice of an appropriate model for object dynamics must be based on a careful study of what an object is likely to do. Perhaps the most popular *tactical target* dynamics model is the *Gauss-Markov Acceleration* model, as proposed by Singer [202]. This model assumes an exponential correlation in time between the accelerations in any direction in inertial space – the state equations for *each* spatial dimension are of the following form:

$$\begin{bmatrix} \dot{p}_{x_{T/I}} \\ \dot{v}_{x_{T/I}} \\ \dot{a}_{x_{T/I}} \end{bmatrix}^i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/\tau \end{bmatrix} \begin{bmatrix} p_{x_{T/I}} \\ v_{x_{T/I}} \\ a_{x_{T/I}} \end{bmatrix}^i + \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix}^i \quad (2.26)$$

where:

$p, v, a$  = (respectively) position, velocity, and acceleration

$x$  = arbitrarily, one of three orthogonal directions in the chosen inertial frame

$T/I$  = indicator that quantities are for the target, relative to the inertial frame

$i$  = indicator that quantities are *coordinatized* in inertial frame coordinates

$\tau$  = correlation time for target accelerations

$w$  = zero-mean white Gaussian noise of strength  $Q$ , as discussed in Sect. 2.3.1

Note that the acceleration in the above equation is the output of a first order lag driven by white Gaussian noise. Simpler models may provide for (1) zero acceleration (constant velocity) targets, or (2) acceleration as a continuous white Gaussian noise process (interpretable as equivalent to the first model but with *pseudonoise* added to allow for filter tuning), or (3) acceleration as the integral of white Gaussian noise (a *Brownian motion* process) of given strength. These modelling possibilities and attendant cautions are discussed from a general but tracking-relevant perspective in [153:180–185]. Some of these and other models, including, for example, piecewise constant (in time) acceleration,

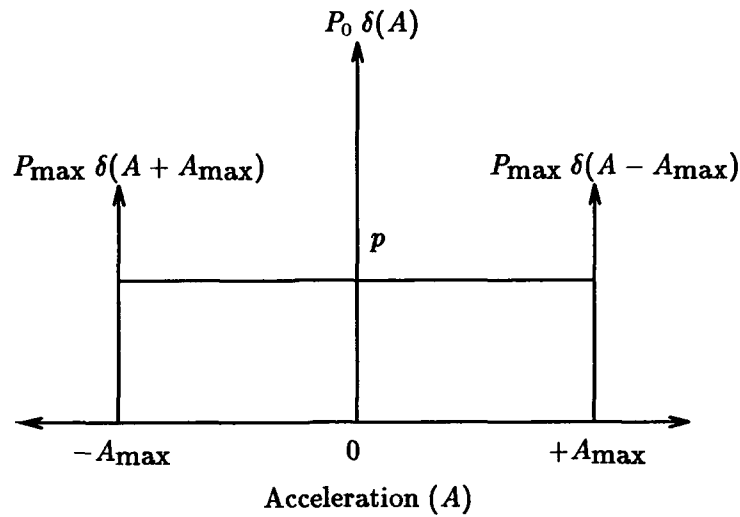


Figure 2.3. The Singer Model for Acceleration Probability Density

with each piecewise constant value found as the realization of a discrete white Gaussian noise process, are discussed in [10:82-88].

Using the Singer approach as in Eqn. (2.26), the designer must specify the correlation time  $\tau$  and the strength  $Q$  (a scalar in any one dimension) of the white Gaussian noise. As discussed in [143], the correlation time is inferred directly from studies of target maneuverability such as [95]. Given a choice for  $\tau$ , and a zero-mean Gaussian approximation for the magnitude of the target acceleration (i.e., a standard deviation  $\sigma$ ), the designer can define  $Q = (2/\tau)\sigma^2$  readily as shown in [153:178].

Recognizing that no real target has a Gaussian acceleration density (since there is no upper bound on the acceleration magnitude for a Gaussian density), Singer [202] defined  $\sigma$  from the second moment of a symmetrical, zero-mean density based on discrete probabilities of some maximum positive or negative acceleration or zero acceleration, with a uniform probability density for accelerations between the maximum and minimum values. This density is shown in Fig. 2.3, where  $A$  represents vehicle acceleration,  $P_{\max}$  is the discrete probability of some maximum acceleration  $A_{\max}$ , and  $P_0$  is the discrete probability of no acceleration. This probability density function and associated formulas are shown in [202, 143, 35, 120].



Another popular dynamics model for tactical targets is the *Constant Turn Rate* model, based on the assumption that the target performs planar, constant turn rate, constant speed, or circular, maneuvers. This model was proposed for ground targets by Burke in 1978 [140:265] and independently for air targets by others around the same time, and investigated for air targets by Maybeck and Worsley [156, 229]. An apparently related target model concept proposed by Bishop [32] is his "*coordinated turn*" model, which provides for an aircraft target to make planar turns with constant lift and "longitudinal" (thrust less drag) force magnitudes.

Finally, we consider measurement noise models. For most radar tracking applications, this issue reduces in practice [35:148] to a selection of measurement noise covariance  $\mathbf{R}$  for discrete time white Gaussian noise added to position and velocity measurements of the form in Eqn. (2.15). It should be noted, however, that radar measurement noise is in fact often *highly* time-correlated, since the noise is a function of the spatial relationships of the tracking radar and the target scatterers [143] [35:161-162] – relationships which are time-correlated by kinematics. One option in this case is to augment the state model with additional noise states as discussed in Maybeck [153:180-185] – the fundamental problem (for a *conventional* radar tracker) here is that, to the extent that the noise state "dynamics" model resembles the target dynamics model, false aimpoint motion from noise and true target motion are indistinguishable (a classic state estimation *observability* problem).

*2.3.2.2 The  $\alpha$ - $\beta$  and  $\alpha$ - $\beta$ - $\gamma$  Filters.* The  $\alpha$ - $\beta$  and  $\alpha$ - $\beta$ - $\gamma$  tracking filters appeared respectively in articles by Benedict and Bordner [24] in the year 1962 and Simpson in 1963 [201]. Neal [171] showed in 1967 that these filters could be interpreted as steady-state Kalman filters. Subsequent developments can be traced in [84, 85, 50, 10].

The  $\alpha$ - $\beta$  and  $\alpha$ - $\beta$ - $\gamma$  filters are distinguished from Kalman filters in that measurements and states in each of two or three dimensions (as required for earth surface or submarine/aerospace tracking, respectively – henceforth we will consider only the three dimensional case) are considered independently, and measurements are provided for position only, corrupted by discrete zero-mean white Gaussian noise. Since the dimensions are considered independently, the nonlinear measurement equation option as used in the ex-

tended Kalman filter (see Eqn. (2.15)) has no counterpart in the the  $\alpha$ - $\beta$  and  $\alpha$ - $\beta$ - $\gamma$  filters. Thus, the  $\alpha$ - $\beta$  and  $\alpha$ - $\beta$ - $\gamma$  filters prescribe independent measurement equations for each spatial coordinate (orthogonal  $x$ - $y$ - $z$  or range-azimuth-elevation) of the form Eqn. (2.6), where  $\mathbf{H} = [1 \ 0 \ 0]$ , and the update equation is written as in Eqn. (2.9), with:

$$\mathbf{K}(t_i) = [\alpha : \frac{\beta}{T} : \frac{\gamma}{T^2}]^T \quad (2.27)$$

where  $T = (t_{i+1} - t_i)$  is the measurement update time interval, and  $\mathbf{K}(t_i)$  is written as a transposed column vector. For particular stochastic driving noise assumptions, explicit equations have been derived for analytical or numerical solutions to  $\alpha$ ,  $\beta$ , and (where applicable)  $\gamma$  (see refs. in previous paragraph).

Considering the hierarchy of choices available for target tracking (the extended Kalman filter, the  $\alpha$ - $\beta$  and  $\alpha$ - $\beta$ - $\gamma$  filters, and finite memory/fading memory filters [154]), Chang and Tabaczynski [50] found the extended Kalman filter in general to be the best choice if computational requirements are not too severe, followed in desirability by the  $\alpha$ - $\beta$  and  $\alpha$ - $\beta$ - $\gamma$  filters for use in cases where computation is severely limited but degraded performance is acceptable. The author does not intend to make use of these filter forms:  $\alpha$ - $\beta$  and  $\alpha$ - $\beta$ - $\gamma$  filters are mentioned here only because they were encountered in related works by other authors, to be discussed later in this chapter.

**2.3.3 Tracking with Kinematic and Signature Measurements.** Developments in this relatively new field have taken two fundamentally different directions – these may be characterized as solving problems in (1) *fire control*, predicting future position of a moving target to provide a gun aiming solution, or (2) *observation-to-track assignment*, as performed in a target acquisition and surveillance system to assign observed targets to existing tracks. Both approaches will be discussed.

The factor common to all techniques discussed in this section is that they fuse kinematic and feature observable information for tracking, or kinematic (principally translation) state estimation. Other tracking techniques using dynamic programming will be discussed in later sections of this chapter. In the next chapter, these existing approaches

can be contrasted to the author's research, which combines kinematic and feature observable information using approaches from this section *and* dynamic programming techniques for object or target recognition.

*2.3.3.1 Fire Control – Approach by Kendrick, Maybeck, and Reid, and Further Developments.* All developments in melding kinematic and feature observable measurements for fire control appear to stem from an observation by Reid [120:iii] in the mid-1970's that target/aspect angle classifiers could help predict the motion of an aircraft target by identifying the orientation of the plane of the wings relative to the velocity vector. Kendrick, Maybeck, and Reid [120, 121] then developed a tracker that employed two coupled Kalman filters: one filter providing a target orientation estimate based on imaging sensor measurements, the other filter providing kinematic state estimates from kinematic measurements and (from the orientation filter) direction (but not magnitude) of "normal load acceleration" (acceleration from lift, normal to the velocity vector).

Subsequent developments of the fire control problem were made almost exclusively by Andrisani *et al.* [5, 4] and Sworder *et al.* [140, 141, 208, 209]. Contributions by Andrisani *et al.* were twofold: (1) an improved dynamics model for conventional aircraft that employed the coordinated turn assumption and included both kinematic states and aspect angles in the extended Kalman filter state equations, producing one filter rather than two coupled ones, and (2) extension of the dynamics and measurement models to the case of a helicopter target. In particular, the Andrisani efforts use image-derived orientation to estimate not only direction of normal load acceleration but also its magnitude. Sworder *et al.* investigated designs for improved tank gunnery against moving ground targets (principally tanks), where the observation by Reid noted above does not apply. Notably, the original estimator form by Kendrick *et al.* continues to be employed in current research, as seen in recent work by Dayton *et al.* [63].

With reference to the discussion in Sect. 2.3.1.2 on acceleration inputs by the aircraft pilot, the motivation behind both the Kendrick and Andrisani-type estimators was really *input estimation* for aerodynamic forces – those forces being most significant with respect to changing the curvature of the trajectory. Differences between the estimated acceleration

and the true acceleration were represented by linear white Gaussian noise-driven processes analogous to those discussed in Sect. 2.3.2. Further development of this concept, apparently not yet pursued, might explore sensor-tracker issues arising from sudden changes in thrust (e.g., changes from cruise to afterburner thrust or back, as revealed by sudden changes in target infrared intensity).

Eagle *et al.* [75] have proposed and investigated a version of the Andrisani estimator using regression dynamics techniques to estimate target parameters for improved tracking and possibly target recognition. In Chapter IV the author will propose a third form of kinematic/aspect-angle filter for use in target recognition, using classical multiple model residual analysis-based parameter estimation techniques [154:129–136] to provide probabilities of target class membership.

Dynamics and measurement equations for the Kendrick and Andrisani estimators are presented in App. C. The form of the Kendrick estimator is illustrated in Fig. 2.4.

The unconventional structure of the Kendrick estimator makes it necessary to provide a short explanation of the sequencing of its operations. Basically, at measurement times, the aspect angle filter is provided with two measurements – a “pose estimate” provided by an image processor, and a “pseudo-measurement” calculated from the propagated target kinematics (discussed in the following paragraph). The aspect angle filter then outputs an estimate of the target aspect angle. Subsequent processing in effect rotates this aspect estimate around the target body pitch axis (nominally, the axis of the wings – see Fig. 5.23) back toward the velocity vector by an angle of attack calculated from the lift magnitude (derived in turn from the kinematic estimate, as in the following paragraph). The resulting orientation of the plane of the wings is taken to indicate the direction of the normal aerodynamic forces (lift, under these assumptions). This direction information is provided to the kinematic filter, which also receives classical range, range rate, angle (i.e., sensor-target pointing angle), and angle rate measurements, as provided by a radar and/or some other sensor suite. The kinematic filter provides an updated kinematic estimate, which is propagated conventionally to the next update time, and the cycle begins again.

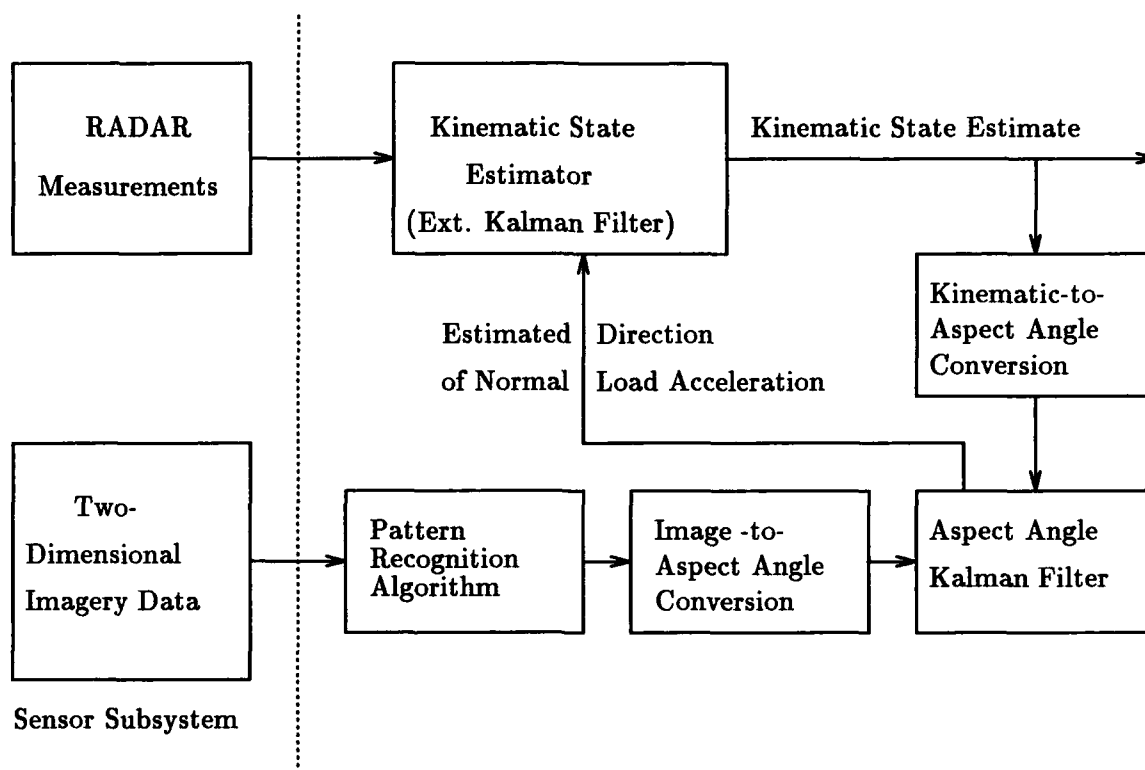


Figure 2.4. The Kendrick-Type Kinematic/Aspect-Angle State Estimator

The pseudo-measurement of target aspect angle is based on the assumption that target acceleration normal to the velocity vector is due only to lift and gravity, and that there is no sideslip (component of velocity normal to the body frame  $\vec{x}_b - \vec{z}_b$  plane : see Fig. 5.23). Target acceleration normal to the velocity vector is available from the filter, the direction and magnitude of gravity are well known, the normal acceleration due to aerodynamic forces is assumed due to lift only, and target mass is assumed known. The standard equation (see Sect. 5.5.3) for lift magnitude as a function of air density, velocity, coefficient of lift, wing surface area (the latter two quantities assumed known for a given target class), and angle of attack then yields the remaining unestimated value – the target's angle of attack. Thus the velocity vector, the direction of normal aerodynamic forces, and the computed angle of attack now completely specify the estimated target orientation relative to the inertial frame, or, with simple coordinate transformation, the pseudo-measurement of target aspect as seen from the attacking aircraft.

Completing the discussion of target models begun in Sect. 2.3.2, we note that the Kendrick estimator uses a dynamics model for normal load (lift) acceleration that, unlike the Singer approach discussed in the earlier section, is biased to provide a higher probability of "upward" acceleration – that is, acceleration due to positive lift, or positive angle of attack. This reflects the realistic desire of pilots to accelerate in a direction they can observe, and to take the resulting acceleration load down into the seat, which is both more acceptable to human physiology and more easily compensated, as by a "G-suit" and/or isometric exercise-like exertions.

This filter system would be expected to provide robust performance for trajectory changes due to rolls, for constant angles of attack. However, because angle of attack is always computed as a function of velocity and acceleration magnitude from the kinematic filter, this system should have problems responding to sudden normal load acceleration magnitude changes resulting from pitch maneuver/angle of attack changes by the pilot. Apparently, consideration of this issue prompted the subsequent development of the Andrisani filter [5].

The form of the Andrisani estimator is conventional, since this estimator consists of one rather than two coupled Kalman filters. The form of the Andrisani estimator is illustrated in Fig. 2.5.

A key factor in both the Kendrick and Andrisani approaches is the assumption that the atmosphere is considered at rest with respect to the inertial frame. This is a significant assumption, since many of the other assumptions (i.e., zero sideslip angle) are in fact wind-relative. To the extent that target-local wind velocity relative to the inertial frame is known, compensation is straightforward [79] – the remaining uncertainty contributes to a form of directionally-dependent bias error falling within the class of errors which should not pose a great problem for the proposed methods.

In contrast to the Kendrick and Andrisani methods, Sworder [140, 141, 208, 209] by necessity took an entirely different approach, although his motivation was much the same. Like Kendrick and Andrisani, Sworder wished to use the imaging sensor basically as an input estimator – estimating the lateral (right-left steering) acceleration input  $u(t)$  for the

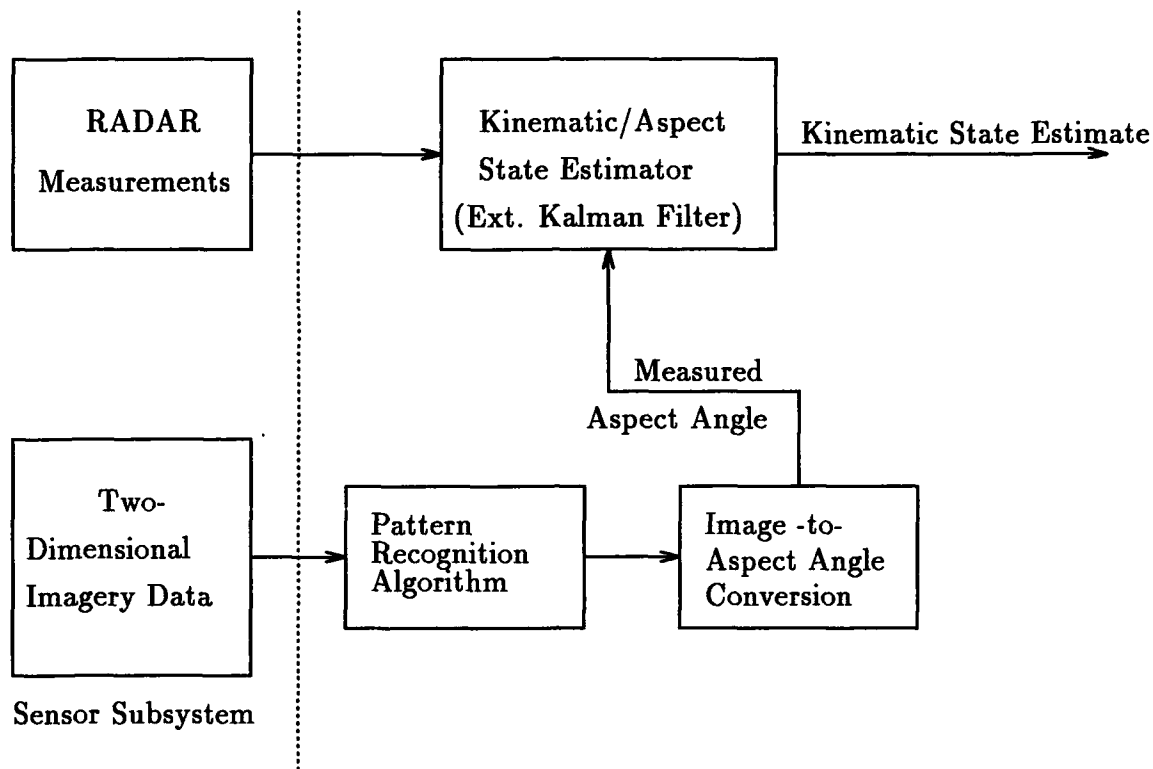


Figure 2.5. The Andrisani-Type Kinematic/Aspect-Angle State Estimator

state equation (Eqn. (2.14)) in an extended Kalman filter which processes also the usual kinematic measurements. However, for the ground targets of Swarder's concern, simple orientation is no longer sufficient to indicate acceleration unambiguously – to estimate lateral acceleration here we must measure the change of orientation over time. Swarder *et al.* attack this problem using the theory of *marked point processes*, as defined by Snyder [205]. Equations relevant to their approach are given in App. C.

It should be noted that use of the Swarder approach for gun lead angle estimation against tank targets may have a practical shortcoming which is not mentioned in the published references [140, 141, 208, 209]. In the case of typical tank-to-tank engagements at ranges of one to three kilometers, where the Swarder estimator ultimately is trying to calculate a lead angle for a tank main gun trajectory solution, it is likely that the target tank also will be attempting to acquire and engage targets of his own. This means that the target tank's turret will be moving, usually under two-axis stabilization, quite

independently in orientation from its hull (i.e., the propulsion system). Since modern tank turrets are so large relative to the hull, and since the hull is more likely to be hidden by terrain, it seems quite likely that *turret* orientation will drive the image-based orientation estimator. Therefore, the use of these orientation estimates for target (*hull*) trajectory estimation could be misleading – in particular if the crew of the target tank *know* that they can confuse enemy fire control by uncoordinated motion of their turret and hull. The key point here is that when the relationship between kinematics and aspect angle breaks down, measuring the latter may not help to estimate, let alone predict, the former.

Concluding this discussion of target trackers fusing kinematic and feature observable measurements, a series of diagrams is presented to show how these systems and their functioning can be represented using the concept of the hypothetical aspect angle sphere, as shown in Fig. 1.2. Figure 2.6 illustrates the target information used and produced by Kendrick *et al.* [120, 121] and subsequently by Andrisani *et al.* [5]. Each circle represents the boundary of a particular 3-D target model sphere, the hypothetical aspect angle sphere centered on the defined centroid of a particular target model. On the spheres are inscribed paths which correspond to aspect angle histories over time. There are fundamentally four different kinds of aspect angle history – true, feature observable-based, kinematic estimate-based, and multi-sensor estimate-based. Marks recorded along the path correspond to particular times at which we synchronize measurements and estimates.

Consider the topmost target model sphere – Model (1) – in Fig. 2.6. The path marked with X's (crosses) is the true aspect angle path – specifically, the path inscribed on the surface of the target model sphere at the point of intersection of the sphere surface and the vector from the center of the target/sphere to the sensor (see Fig. 1.2). The set of (true) feature observable values corresponding to any point along this path can be described uniquely as a function of azimuth and elevation angles relative to a target-model body coordinate frame. The angular “roll” dimension about the vector is unimportant here because we consider only (in-plane) rotation-invariant feature observables (should this target/sensor-relative “roll” become critical for a particular sensor, it could be considered in defining true or expected signatures). In any given engagement, this true path defines the set of feature observables which would be observed with perfect measurements. Note



Fig. 2.6.a

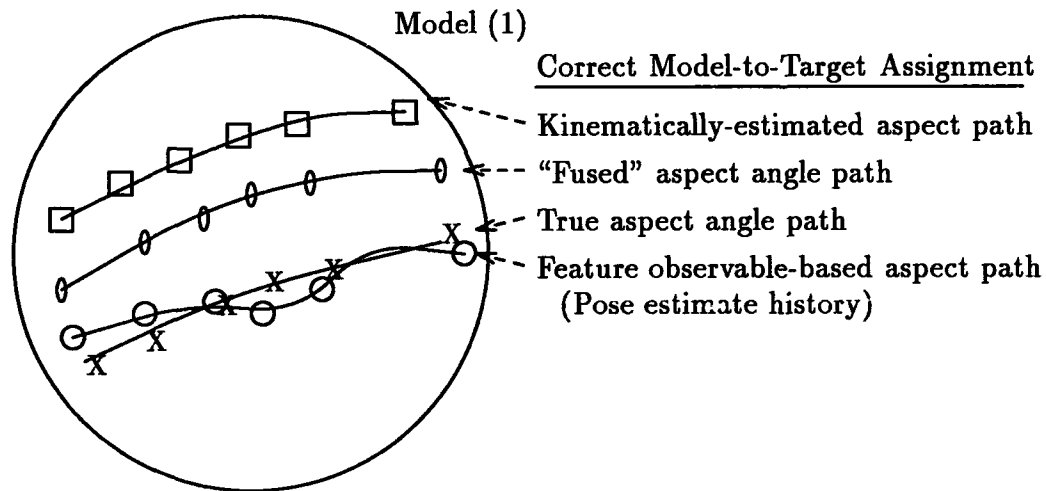


Fig. 2.6.b

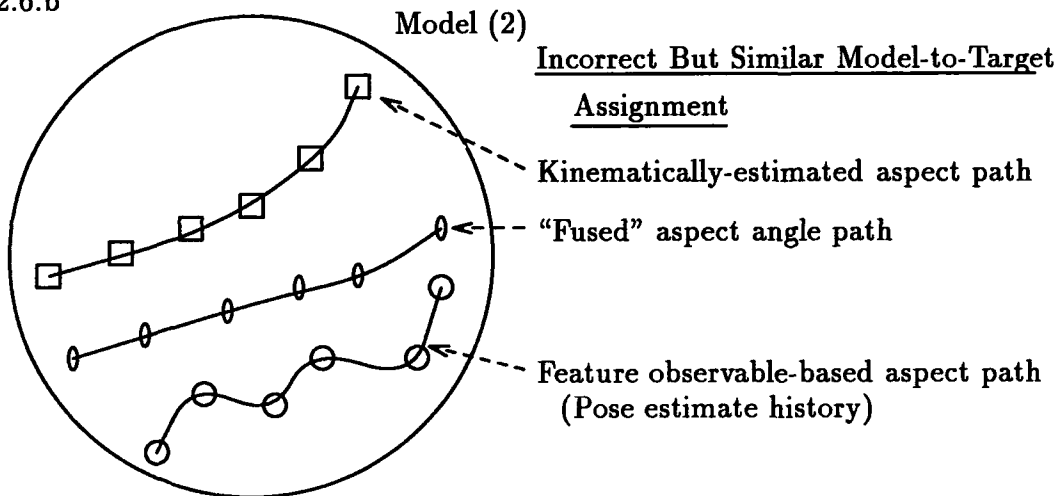


Fig. 2.6.c

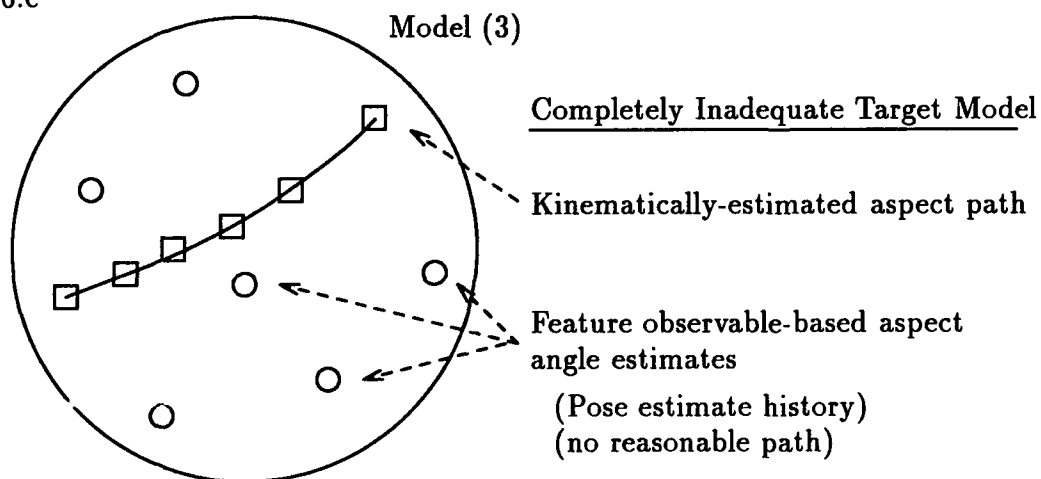


Figure 2.6. Aspect Angle Information Fusion in the Kendrick-Type Estimators

that, in any single engagement (one target), there is only one true aspect angle path – corresponding to the true model out of (say)  $J$  possible models.

The path marked with o's (circles) is the feature observable-based path, or pose estimate history. This is the aspect angle path that would be calculated (estimated) for any given target model using the set of corrupted measurements extracted by viewing the true model over the true path with an actual sensor. As discussed in Sect. 2.2.1, for any given measured feature observable value, the corresponding aspect angle or pose estimate is extracted from each model or associated signature library, generally using nearest neighbor or lookup techniques. Probabilistically, this aspect angle (state) estimate in general represents  $\arg_{\mathbf{x}^a} \{\max[p(\mathbf{z}^f | \mathbf{x}^a)]\}$ , that is, a maximum (classical) likelihood estimate of the aspect angle  $\mathbf{x}^a$  for some feature observable measurement  $\mathbf{z}^f$ , where the superscript  $f$  denotes "feature". Aspect angle estimates from kinematics would be used at most only to start the search in some acceptable local neighborhood – to prevent the algorithm from defining an obviously incorrect aspect angle, often due for example to planes of symmetry which give nearly identical feature values for aspect angles differing by 180 degrees.

Note that, if the (true) target model from which the measurement is extracted is identical to the (test) target model from which the corresponding aspect angle is sought, then there are only three reasons for an incorrect aspect angle determination: (1) production of ambiguous (ill-defined, as for the visual image of a sphere, or noisy, as for radar scatterer interactions) signatures at the target, (2) transmission noise, and (3) sensor noise. Thus, for low-ambiguity signatures and low transmission/sensor noise, the true aspect angle path and the feature observable-based aspect angle path on the correct model should not differ greatly. However, if the true model and the test model differ, that model mismatch provides an additional source for error in the feature observable-based aspect angle path, as represented in the center diagram, Model (2), of Fig. 2.6. Finally, one can envision cases (e.g., Model (3) in Fig. 2.6) in which severe mismatch between true and test models results in a feature observable-based path that resembles a random walk over the surface of the target sphere, perhaps not a reasonable aspect angle path in any sense. Clearly, severe measurement-to-test model mismatches may provide grounds for

terminating determination of a feature observable-based aspect angle path early in the process.

The path marked with squares is the kinematics-based aspect angle path. This is the aspect angle path that would be expected, given kinematic estimates as provided by a sensor/estimator suite measuring kinematic variables only (principally position and velocity), with additional information on dynamic limitations of each target model. In the aircraft tracking work discussed here, these dynamic relations are the basic coordinated turn, lift (lift normal to velocity vector, nearly normal to wings), and angle of attack (alpha nearly proportional to magnitude of lift, inversely proportional to speed squared) assumptions found in Kendrick *et al.* [120:11] and elaborated by Andrisani *et al.* [5].

The path marked with ellipses is the multi-sensor estimate-based ("fused") aspect angle path – an aspect angle path based on some optimal estimator using both kinematic and feature observable measurements. If we use the correct model as in the top diagram in Fig. 2.6, it would be reasonable to assume that the fused path will fall between the kinematic path and the feature observable path, or, for small (or at least unbiased) sensor errors as shown, closer to the true aspect angle path than we could achieve with kinematic estimates alone. But if we choose the wrong target model, the fused aspect path could easily be a poorer estimate than provided by the kinematic path alone. Thus, the Kendrick estimator and its derivatives should be very sensitive to proper choice of target model – the research demonstrated in Chapter IV exploits this sensitivity.

The preceding discussion spans the extent of solutions for the fire control problem by fusing kinematic and feature observable information. Developments discussed below by authors other than Kendrick *et al.*, Andrisani *et al.*, and Sworder *et al.* have been directed instead toward observation-to-track fusion.

**2.3.3.2 Observation-to-Track Assignment: Approach by Bar-Shalom.** Bar-Shalom's probabilistic technique for observation-to-track fusion, called **Joint Probabilistic Data Association (JPDA)** in its latest form, is a method for updating a set of kinematic track files where several observations can be associated with any given track [33:299–304] [10, 8]. Fundamentally, the method assigns to each observation within some predic-

tion "gate" (i.e., a region in space around the nominal predicted position for the next observation) a probability that *that* observation is the correct one. This probability is a function of the distance of the observation from the anticipated new observation position, as based on propagation of the track state vector from the last update to the current time. Finally, the existing track is updated by a "pseudo-residual" constructed as a linear combination of all the residuals (differences between each observed possible position and the predicted position) in the prediction gate, each weighted by its probability of association to the existing track.

Bar-Shalom's writings discuss this fact only briefly [10:229], but the JPDA structure is inherently well suited to incorporating information from feature observable measurements. Instead of, or in addition to, weighting the probability of correct association based on spatial distance from each observation to the predicted nominal new location, one could weight each observation probabilistically based on its distance from the previous track in *feature (observable) space*.

**2.3.3.3 Approach by Blackman.** Blackman's approach to fusion of kinematics and "attributes" [8:205-209] [33:376-380] is apparently derived from the approach taken by Bar-Shalom, and represents one manner in which Bar-Shalom's recommendations could be implemented. Blackman's approach is summarized by the equation below (shown here exactly as it appears in [8], except that Blackman uses a capital **Z**, which in the notation to be introduced in the following section will mean a *measurement history* of several measurements over time, rather than one measurement, as Blackman intended):

$$\mathbf{P}_{ij} = \left( \frac{\mathbf{P}_D \exp(-\mathbf{d}_{ij}^2/2)}{(2\pi)^{M_{ij}/2} \sqrt{|\mathbf{S}_{ij}|}} \right) \left( \frac{1}{\tilde{\mathbf{V}}_{ij}} \right) (\mathbf{P}(\mathbf{z}_j | \hat{\mathbf{z}}_i)) \quad (2.28)$$

in which  $\mathbf{P}_{ij}$  is the probability that observation  $j$  belongs to track  $i$ ,  $\mathbf{P}_D$  is the probability of detection,  $\mathbf{d}_{ij}$  is the quadratically-weighted distance (i.e.,  $\mathbf{d}_{ij}^2 = \mathbf{t}_{ij}^T \mathbf{S}_{ij}^{-1} \mathbf{t}_{ij}$ , where  $\mathbf{t}_{ij}$  is a measurement residual vector) from observation  $j$  to the predicted nominal location for the next observation of track  $i$  in  $M_{ij}$ -common dimensional space ( $M_{ij}$  is the dimension of the intersection of the measurement and track spaces), and  $\mathbf{S}_{ij}$  is the covariance of the estimate

for the predicted nominal.  $\tilde{V}_{ij}$  is a "measurement volume" difference parameter attempting to quantify the magnitude of the mismatch in spatial dimension and size between (1) the space observed by the sensor taking measurement  $j$  and (2) the space traversed by track  $i$  to be updated (for example, quoting Blackman [8:205-206], "given an angle-only track and a radar measurement that includes angle, range, and range rate,  $\tilde{V}_{ij}$  is the extent of the range and range-rate measurement space"). Finally,  $P(z_j | \hat{z}_i)$  is an "attribute likelihood function," giving the probability that the feature observables for the new observation could have the values  $z_j$  measured for observation  $j$ , given the predicted feature observable values  $\hat{z}_i$  for track  $i$ .

Although it appears that Blackman intended to use  $P_{ij}$  as a weighting factor in the Bar-Shalom fashion, it should be noted that this quantity could be used to define a "nearest neighbor" or "maximum likelihood" association technique as well, assigning the single observation  $j$  to that track  $i$  for which  $P_{ij}$  is greatest. In any case, Blackman's approach appears highly heuristic: it is related to, but does not appear to be explicitly derivable from, maximum likelihood methods.

*2.3.3.4 Approach by Mitzel.* Whereas Bar-Shalom and Blackman took a probabilistic or statistical approach to observation-to-track fusion, fusing information from multiple observations into one track, Mitzel's approach [8:297-320] is a classic nearest neighbor approach. Mitzel defined an augmented vector consisting of four kinematic states - position and velocity in each of 2 dimensions - followed by a set of  $m$  global descriptor feature observable quantities. The kinematic states are updated in an  $\alpha - \beta$  tracker, but the feature observable quantities are treated as constant with no driving noise, and updated using the simple "static estimator" form of the linear Kalman filter [153:9-15], under the assumption that the feature observable measurements are corrupted by discrete-time white Gaussian noise.

For each observation in a track gate defined by the kinematic estimate covariance, Mitzel defines an augmented vector of the same form as that maintained for the track itself. Propagating the track forward to the measurement time, Mitzel then simply determines the "nearest neighbor" observation to the propagated track vector. His distance metric

is Euclidean in the  $4 + m$  dimensional augmented vector space, weighted only by the covariance of the vector elements (due to the dimensional independence assumptions in the  $\alpha - \beta$  tracker structure and and Mitzel's assumed independence of feature observables and kinematics, the only non-zero off-diagonal covariance elements are those relating kinematic states in any one dimension).

The implicit assumption of independence between kinematics and feature observables made by Mitzel was viewed by the author as a significant limitation to the performance of this algorithm. The original motivation for the start of this research was a desire to improve upon the assumptions in the Mitzel effort.

*2.3.4 Object Tracking / State Estimation - Conclusion.* With the end of this section, this chapter has covered nearly all of the previous or classical approaches to physical object or tactical target recognition and tracking. From this point on, we will introduce ideas that are new, or at least unconventional, with respect to these classical approaches. The eventual intent is to apply these new ideas to dynamic object recognition, in particular for moving objects.

## *2.4 Sequence Comparison by Dynamic Programming*

This section introduces the mathematics of sequence comparison by dynamic programming methods. A central focus of this research is the use of dynamic programming to compare (1) measured sequences of feature observables from an unclassified object against (2) candidate sequences or regions from potential or known object classes. The output of this comparison process will be a set of likelihood function values, or measures of the likelihood that the observed sequence was generated by each of the candidate objects.

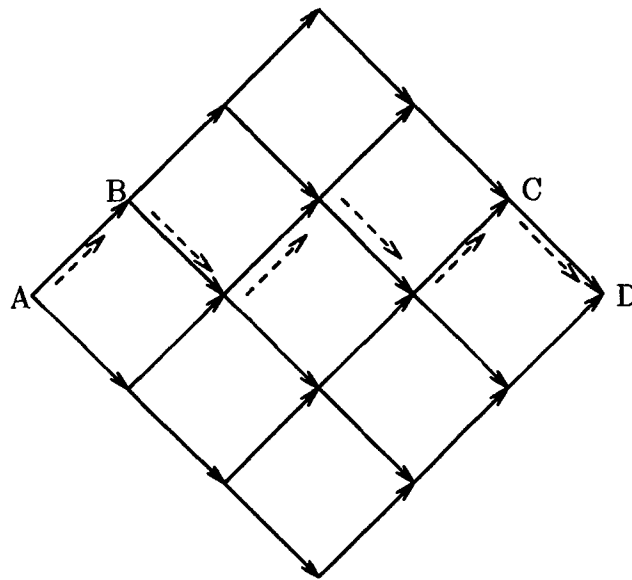
Recalling the discussion on "independent look" pose estimates in Sect. 2.2.1, we intend to show that dynamic programming sequence comparison methods offer a natural method to *restrict* pose estimate histories to reasonable aspect angle progressions, in accordance with other information about candidate object classes. In the following section, we will see why this restriction improves object recognition.

**2.4.1 Dynamic Programming.** Dynamic programming (DP), sometimes called the "*shortest path algorithm*," is a technique developed by Bellman [23] for making optimal decisions with minimum effort, where we define optimal as "minimum cost," according to some specified cost criterion, or "minimum *expected* cost," where stochastic effects result in some non-zero probability that our decisions will not be implemented as we intend [155, 71, 67]. In practical use, dynamic programming is often presented as a tool to find a minimum cost (length) path through a sequence of discrete states at discrete times, as shown in Fig. 2.7, where each "branch" in the lattice represents a state transition carrying a given cost, and the total cost of a path is the sum of the cost of the individual steps.

It is important to note here that the term "state" carries a distinctly different connotation in dynamic programming than in general estimation and control practice. A "state" in DP is generally a discrete location or point in some generally continuous, multidimensional state space, rather than a particular *dimension* in that state space, as is the usual meaning in estimation and control. This distinction arises because DP is only practical where a continuous state space can be discretized to define a finite (therefore countable) number of locations.

The inelegant (maximum effort) method of solving the problem presented in Fig. 2.7 would be by *exhaustive search*, i.e., to determine the cost of every possible path through the lattice, and select the path of least cost. However, if our system conforms to Bellman's "*Principle of Optimality*," then dynamic programming may allow us to determine the minimum cost path without considering every possible path. For *backward* dynamic programming, a classical tool in control applications where we wish to determine the minimum-cost sequence of state transition decisions required to reach a given final point from a given start point, the *Principle of Optimality* states that (from [71:3], with modification to emphasize that this is a property that a system *must be known or assumed to have* in order to use dynamic programming, but *may not have in actuality*):

The best (minimum cost) path from A to D *must have* the property that, whatever the initial decision at A, the remaining path to D, starting from the next point B after A, must be the best path from that point to D.



Note: Each path step has an associated cost.  
Dotted arrows mark minimum cost path  
from A to D.  
Diagram taken from [71] with  
modification.

Figure 2.7. A Minimum Cost Path Defined By Dynamic Programming



If this is true, the second (and last) key idea of backward dynamic programming is that the optimal path/control sequence is solved by working backward from the desired endpoint to the beginning, recording the minimum cost decision for each state as we go (this recording process is often referred to as the setting of "*pointers*"). At the beginning point, we then retrace our steps through the *stages* (generally, state/time combinations) back to the end to find the optimal path.

A straightforward extension of backward dynamic programming leads to *forward* dynamic programming [71:10-11], the form commonly used in sequence comparison. For forward dynamic programming, executed in the reverse direction with respect to the previous paragraph, the *Principle of Optimality* states (with reference again to Fig. 2.7):

The best (minimum cost) path from A to D *must have* the property that, for any point C along that path before D, the best path from A to C must lie along that best path from A to D.

Using these rules, the challenge in using dynamic programming for any particular application is to determine the incremental costs, transition rules (since not all paths or transitions may be allowed), and (for stochastic problems) transition probabilities – i.e., the probability that a given transition will be executed, given that *that* transition or some other one was chosen. Limiting the usefulness of dynamic programming, however, is the "*Curse of Dimensionality*," [67:41, 75-76] – as the number of possible states and/or times grows, the number of possible paths (dimensionality) grows explosively. Although dynamic programming requires the consideration of fewer paths than exhaustive search, there is still a price to be paid, and that price can become prohibitive.

**2.4.2 Dynamic Time Warping for Speech Recognition.** The technique of *dynamic time warping* (DTW) was born from the observation that the utterance of spoken words is a stochastic process – no two persons pronounce the same word identically, and, moreover, the same person does not pronounce any given word exactly the same on any two occasions, due to a variety of factors (level of stress, word context, etc.). It was further reasoned that a considerable portion of the difference between any two utterances or realizations of a given word could be charged to stretching (expansion) or compression of portions of one utterance relative to the other, or possibly to addition or deletion of relatively minor

sounds. If the effects of these differences could be eliminated, or "*time warped*," so the idea went, the underlying similarity of the two words could be revealed.

With this objective, the concept of using dynamic programming for time warping and, by extension, speech recognition, first appeared in the open press in 1970-71 in articles by Velichko [214] and Sakoe and Chiba [192]. Subsequent research in this field was conducted and published by Itakura [115], White and Neeley [223], Sakoe and Chiba [193], and (leading efforts in this country) Rabiner *et al.* at Bell Laboratories [182, 183, 168]. Texts by Parsons [176] and Sankoff and Kruskal [195] are useful references.

Explanations of dynamic time warping and dynamic programming-based sequence comparisons in general require the use of a diagram like Fig. 2.8. Each axis of the diagram represents a sequence of twelve elements that is to be compared to the other sequence of twelve elements – the circles represent possible (but perhaps forbidden) associations between the elements. The nature of the elements is completely general – often they are vectors of some form, but all that we will require is that some distance metric exists by which one element can be compared to another.

Often, each element represents a discrete, sampled data representation from some feature space. The features generally represent observable quantities due to a physical (i.e., classically continuous) process or trajectory in some state space, where the true location in the state space at any time is unknown – the distinction between the *state* space of this trajectory and the *feature* space of the observables is an important one, and not always clear in the literature (see discussion in Chapter I of [195]). Note that, in general, the two sequences need not contain an equal number of points.

The "warping" or sequence comparison is really a process of making associations between individual elements in the two sequences, computing the cost of each association according to the *distance* (measure of dissimilarity) between the element in one sequence and the element in the other, and finding the set of associations or *matching path* that gives the minimum total cost or distance. Associations are made subject to "continuity constraints," that limit, for example, the number of associations that can be made from one element of one sequence to elements of the other sequence, the number of elements

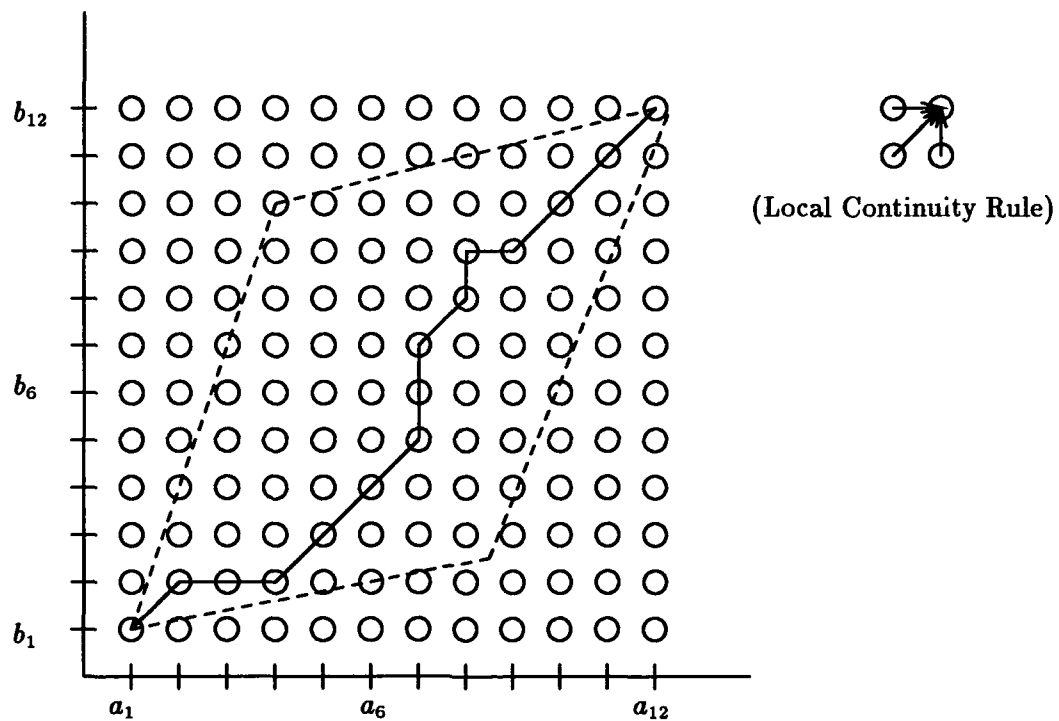


Figure 2.8. Classical Dynamic Time Warping

in a sequence that can be skipped (if any), matching paths that move to the left or down (generally, these paths are *prohibited*), and so on. Continuity constraints prevent undesirable low cost associations between two sequences that really have significant differences, and they lower the risk of incurring the *Curse of Dimensionality* (see previous subsection).

For example, in Fig. 2.8, the dotted lines represent *global* continuity constraints, or bounds on the region of admissible paths – we have chosen not to allow or consider, for example, association of element  $a_{10}$  with element  $b_2$ . As an example of an undesirable low cost association path, suppose that *all* of the elements in sequence  $B = \{b_1, b_2, b_3, \dots, b_{12}\}$  are “closer” or more similar to element  $a_1$  than to any other element in  $A = \{a_1, a_2, a_3, \dots, a_{12}\}$  in some chosen metric. This means that the (unconstrained) association path between  $A$  and  $B$  will be a vertical line segment through the leftmost column of circles. But suppose that the analyst is absolutely sure from some other source of information that  $a_1$  cannot possibly correspond to, say,  $b_3$  and above – continuity constraints can prevent this

“impossible” match and force the system to output a warping path cost which is more representative of (if not explicitly giving) the probability that sequences *A* and *B* arise from the same underlying space.

Conversely, the global constraints in Fig. 2.8 show that we have chosen to *require* the association of the first and last elements of each sequence with the first and last elements, respectively, of the other. The *local* continuity constraints for this diagram are shown in the extracted portion on the right – the arrows indicate that allowable paths into any association cell can only come from the left, lower left, or lower cell. This simple local constraint is the most common of many alternatives [195:136].

A final form of path constraint for which numerous alternatives have been proposed is path distance weighting. The reader will observe that a straight (diagonal) path through the array in Fig. 2.8 will contain fewer elements, and therefore a lower total path cost if this total cost is a simple sum of individual association costs. For some applications, the designer may wish to relax this preference for diagonal paths, and may provide de-weighting or “edge-weighting” techniques to make non-diagonal paths more feasible. For example, considering the simple local constraint discussed above, one may choose to weight costs from horizontal and vertical transitions by a multiplicative factor of  $\frac{1}{2}$ , while costs from diagonal transitions are unweighted. This form of weighting (discussed in [195]) tends to make warping paths that move horizontally, then vertically (or vice-versa) just as feasible as those that move diagonally.

In Fig. 2.8, then, the solid line represents the minimum cost sequence of associations between the elements of the two sequences *A* and *B*, subject to these global constraints, and possibly other constraints that need not be specified explicitly for this example. This is a minimum cost path through the space of associations, subject to transition constraints, and forward dynamic programming provides a natural approach to determine this path. Note that backward dynamic programming would be equally applicable, but researchers in this field prefer to start associations from the beginning rather than from the end – working forward allows one to attempt to work in real time, and word beginnings are also generally more distinctly spoken and identifiable than word endings.

Thus, for the simplest form of local continuity constraint, allowing diagonal, vertical, or horizontal associations, but no additions or deletions (skipping an element of one or the other sequence) as shown in Fig. 2.8, the forward dynamic programming equations can be written and executed from left to right as:

$$D(C_k) = d(c_k) + \text{MIN}[D(C_{k-1})] \quad (2.29)$$

where:

$c_k = [a_j, b_l]$  is the  $k$ -th element in a sequence of allowable associations of elements from sequence  $A$  with elements of sequence  $B$ , this particular association being between element  $a_j$  and element  $b_l$

$C_k = \{c_1, c_2, c_3, \dots, c_k\}$ , the minimum cost sequence of associations leading to and including association  $c_k$

$d(c_k)$  = the cost or distance of association  $c_k$ , i.e., the distance in some metric between element  $a_j$  and element  $b_l$

$D(C_k)$  = the total cost of reaching and accomplishing association  $c_k$  by the minimum cost sequence of allowable associations

It is important to point out that, for sequence comparison in general and speech recognition in particular, there is no body of theory prescribing optimum feature space representations, distance metrics, or path constraints. These choices have been the subject of much experimental research, without identification of one particular "best" approach [176:297-303] [223:186-187] [168:634] [195:125-161], although various researchers have preferences for certain approaches in particular cases, e.g. [115] [195:37-40].

The key point here is that, in contrast to dynamic time warping, for *motion warping*, or the application of dynamic programming sequence comparison in dynamic object recognition, the dynamic state restrictions of each object class can determine *natural relationships* that provide for continuity constraints to be defined analytically. *These continuity constraints are just exactly the restrictions on pose estimate transitions that were motivated in Sect. 2.2.1.*

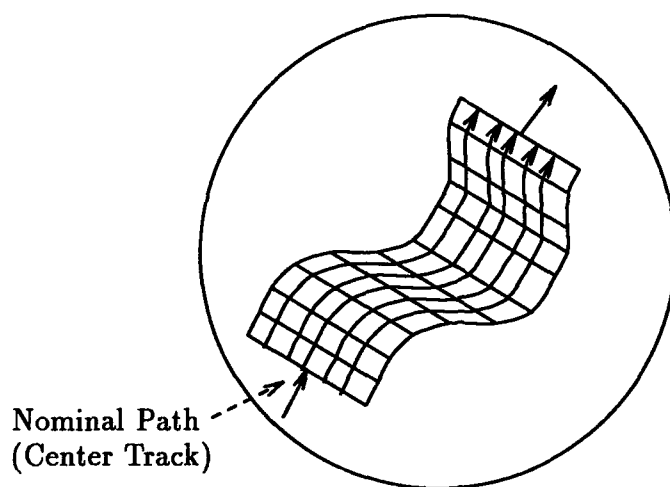
**2.4.3 General Sequence Comparisons.** Apart from application in speech processing, dynamic programming-based sequence comparisons have seen use (and independent origins) in a wide variety of fields, as discussed in [195]. These fields include chromosome comparison, dendochronology (archaeological dating by tree ring analysis), bird song comparison, recognition of partial images (by Gorman *et al.* [101], to be discussed in the following chapter) and so on.

An extensive literature search, however (see bibliography), has failed to find any use of such *classical* sequence comparison techniques for recognition of moving (changing aspect angle) objects. Telephone conversations with Gorman [100] and Rabiner [180] support this observation. In Chapter III, it will be shown that some published dynamic programming-based techniques for trajectory estimation [129, 66] and object recognition [136, 164, 165] can be viewed as sequence comparison techniques. This research serves in part to fold these and other methods into the larger class of algorithms for sequence comparison by dynamic programming. Application of the classical sequence comparison techniques discussed above and other aspects of this research will considerably widen the range of approaches for dynamic object recognition – any of these sequence methods can provide significant performance improvements over the conventional “independent look” methods discussed in Sect. 2.2.1.

In this research, we will use the term “warping path region” or “space” when referring to the finite set of all possible associations between elements of two finite sequences of feature observable vectors, from which sets of associations may be defined under applicable rules to define warping paths. This warping path space is illustrated in Fig. 2.8, for warping of two one-dimensional sequences (producing a “two-dimensional” warping path region or space).

In a key generalization to basic DTW sequence comparison, we may conceive of the need to match a one-dimensional measured sequence with a *two-dimensional* origin region of possible sequences on some model. Fig. 2.9 illustrates the *three-dimensional* space of allowable matching paths that results from this matching of a one-dimensional sequence with allowable paths through a two (angular) dimensional aspect angle region, as opposed to the *two-dimensional* space of paths portrayed in Fig. 2.8.

Fig. 2.9.a: Target Model Sphere



Warping Path Allows Off-Nominal Matches

Fig. 2.9.b: Three-Dimensional Warping Path Diagram

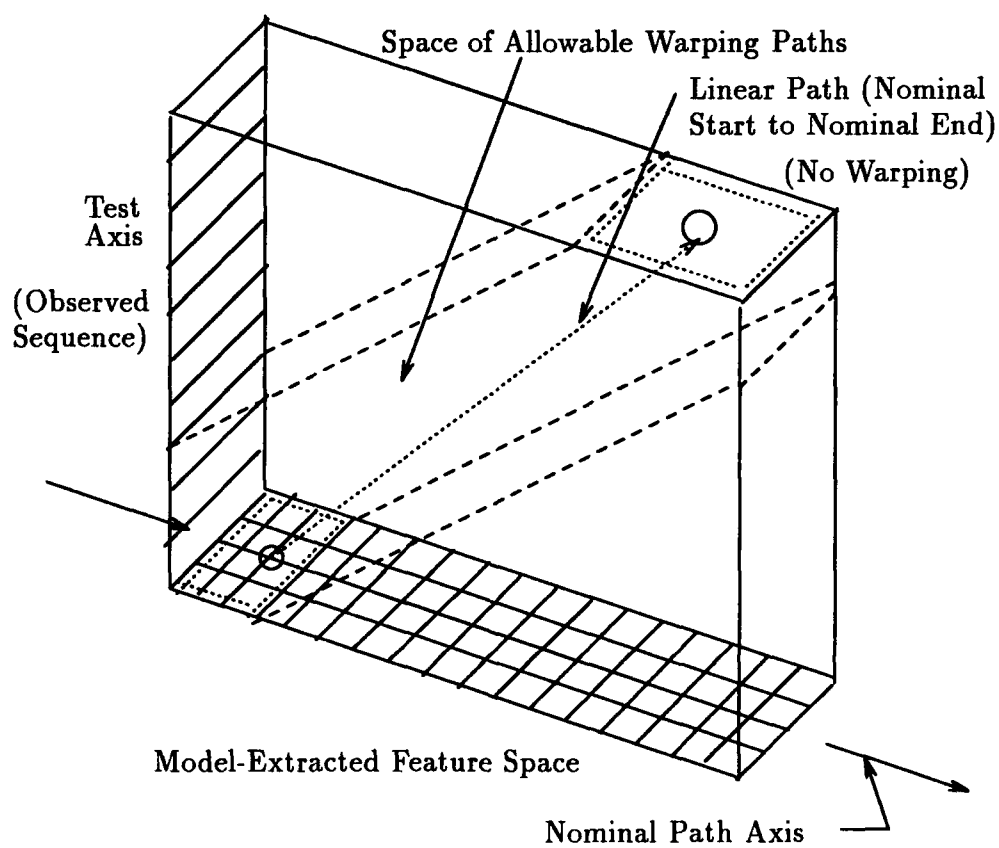


Figure 2.9. Motion Warping in a Two-Dimensional Aspect Angle Region

The one problem with this approach is that we are more likely to run afoul of the "Curse of Dimensionality" [23, 71, 155] by allowing for a "three-dimensional" space of paths. Otherwise, the extension to this technique, using somewhat more elaborate three-dimensional continuity constraints, will be straightforward. The continuity constraints for this approach – the width of the allowable path region, lateral (as opposed to along the nominal direction) path transition rules, and so on, can be defined similarly to those discussed above.

Note that each point in warping path space is really an association between two points in the feature observable space. These feature observable values may arise from different points in the underlying state space (which will be aspect angle, in our case), but the association cost is based only on the difference in feature observable values. The continuity constraints of classical sequence comparison methods may modify these association costs to deter or prevent matches that appear to be too far away in a state space sense (i.e., separated by too many sequence elements). However, the sequence matching cost is not *explicitly* weighted by any factor arising from other knowledge that may be available about transitions in the state space which produces the measurements. To do that, we will apply the following dynamic programming state estimation algorithm – a unique form of dynamic programming sequence comparison, but one not heretofore associated with the family of *classical* DP sequence comparison techniques.

**2.4.4 The Larson and Peschon (L&P) Algorithm.** Larson and Peschon (L&P) proposed a dynamic programming-based algorithm [133] for estimating the sequence of  $n$  states or locations in some space with maximum *a posteriori* or MAP probability of producing an observed sequence of  $n$  measurements, conditioned on *a priori* information about the likelihood of transitions in the state space. They did not motivate their work as a tool for object recognition working on an aspect angle space, but we will ultimately apply it in this fashion. We will discuss this algorithm, and later illuminate the relationships between it and the classical dynamic programming sequence comparison techniques discussed above.



Given a sequence of measurements  $\mathbf{Z}_k = \{z_1, z_2, \dots, z_k\}$ , Larson and Peschon desired to find the sequence of states  $\hat{\mathbf{X}}_{k/k} = \{\hat{x}_{0/k}, \hat{x}_{1/k}, \hat{x}_{2/k}, \dots, \hat{x}_{k/k}\}$  that maximized the conditional probability density function:

$$p(\hat{\mathbf{X}}_{k/k} | \mathbf{Z}_k) = \underset{\mathbf{X}_k}{\text{MAX}} [p(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k | z_1, \dots, z_k)] = \underset{\mathbf{X}_k}{\text{MAX}} [p(\mathbf{X}_k | \mathbf{Z}_k)] \quad (2.30)$$

where the term "MAX" refers to the operation of finding the maximum value of the indicated term, over all values of  $\mathbf{X}_k$ , representing the sequence of states  $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ . The *a priori* information as to the likelihood of transitions in the state space, as discussed above, is represented as  $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$  for any choice of  $k$ . Note, as do Larson and Peschon, that the intent here is to estimate the entire sequence up to the present, rather than simply the present state  $\mathbf{x}_k$ .

Next, Larson and Peschon were willing to assume independence of measurements  $z_j$  from states  $\mathbf{x}_i$  and measurements  $z_i$  for  $t_j \neq t_i$ , implying, for example, that the time interval required to take data for one measurement is less than, and synchronized with, the loiter time in any one state, and that the measurement instrument is independent from event to event. These assumptions are typical of discretization assumptions made to implement dynamic programming algorithms in naturally continuous spaces, or to limit the dimensionality of a problem in naturally discrete spaces (where a "coarser" discretization than the natural level may be assumed to reduce computation).

These assumptions appear to be reasonable for object recognition algorithms in which measurements arise from an aspect angle state space and sensors have high "bandwidth" relative to the observable state transition processes. In other words, we expect the assumptions to be appropriate where the sensors have shorter response times than the time intervals over which the underlying state space is expected to change enough to alter the "mean" signature – effects on signatures due to state changes over the time required to make a measurement will be indistinguishable from the effects due to noise if there were no state change. In Chapter III, we will discuss a particular case of interest to us in which

these conditions may not apply, and an approach by which the Larson and Peschon assumptions might be relaxed to address this issue. It will be clear that the assumptions made by Larson and Peschon above *can* be relaxed if required without destroying the utility of their basic approach, but at the cost of increased dimensionality, in that probabilities or likelihoods (costs, in the general dynamic programming sense) become *path and/or measurement dependent* – this is what Larson and Peschon are trying to avoid through their assumptions.

The extent to which discretization assumptions are reasonable or restrictive in any particular scenario, however, must be evaluated on a case-by-case basis. This evaluation can be conducted from first principles – assessing, for example as we did in the previous paragraph, whether or not in any particular state transition / measurement scenario, assumptions like those of Larson and Peschon are reasonable. Alternatively in test and ultimately in practice, the evaluation will be done empirically – obtaining a problem solution (a control law, a target recognition algorithm, etc.) under the discretization assumptions, and assessing whether or not that solution provides acceptable performance when applied to states and/or measurements from a continuous or more finely discretized “truth” scenario. In general, as in sampling problems, the assumption of discretization becomes more physically reasonable (if not more feasible computationally) as the discretization fineness increases. Where an overall fine level of discretization leads to high dimensionality and the “curse of dimensionality” becomes an issue, it may be reasonable to limit high dimensionality or fine discretization to particular subsets of the space where the optimal answer is expected to lie, perhaps using iterative solution processes to converge to an adequate answer [155:239, 247, 256–257].

In any case, their assumptions allow Larson and Peschon to use Bayes’ Rule and break the maximization process into stages, making it suitable for solution by dynamic programming with the following equations (use of which will be discussed below):

$$p(\hat{X}_{k/k} | Z_k) = \text{MAX}_{\mathbf{x}_k} \left\{ \text{MAX}_{\mathbf{X}_{k-1}} \left[ \frac{p(\mathbf{X}_k, Z_k)}{p(Z_k)} \right] \right\}$$

$$= \text{MAX}_{\mathbf{x}_k} \left\{ \text{MAX}_{\mathbf{X}_{k-1}} [p(\mathbf{X}_k | \mathbf{Z}_k)] \right\} = \text{MAX}_{\mathbf{x}_k} I(\mathbf{x}_k, k) \quad (2.31)$$

which shows the final step in the process, a maximization of  $I(\mathbf{x}_k, k)$  over all possible final states  $\mathbf{x}_k$ , where:

$$\frac{p(\mathbf{X}_k, \mathbf{Z}_k)}{p(\mathbf{Z}_k)} = p(\mathbf{X}_k | \mathbf{Z}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k)p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p(\mathbf{z}_k | \mathbf{Z}_{k-1})} \frac{p(\mathbf{X}_{k-1}, \mathbf{Z}_{k-1})}{p(\mathbf{Z}_{k-1})} \quad (2.32)$$

and:

$$I(\mathbf{x}_k, k) = \text{MAX}_{\mathbf{X}_{k-1}} [p(\mathbf{X}_k | \mathbf{Z}_k)] \quad (2.33)$$

Then, stepping theoretically to a hypothetical  $k + 1$ -st step:

$$I(\mathbf{x}_{k+1}, k + 1) = \text{MAX}_{\mathbf{X}_k} \left[ \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})p(\mathbf{x}_{k+1} | \mathbf{x}_k)}{p(\mathbf{z}_{k+1} | \mathbf{Z}_k)} p(\mathbf{X}_k | \mathbf{Z}_k) \right] \quad (2.34)$$

or, equivalently, in the recursive form which is the heart of the algorithm:

$$I(\mathbf{x}_{k+1}, k + 1) = \text{MAX}_{\mathbf{x}_k} \left[ \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})p(\mathbf{x}_{k+1} | \mathbf{x}_k)}{p(\mathbf{z}_{k+1} | \mathbf{Z}_k)} I(\mathbf{x}_k, k) \right] \quad (2.35)$$

Since the factor  $p(\mathbf{z}_{k+1} | \mathbf{Z}_k)$  is the same for all maximizations made at any time  $t_k$ , the actual maximization at any stage need not be done over the term shown in brackets in Eqn. (2.35), but rather only over the expression defined by computing this term without its denominator, denoted  $I^*(\mathbf{x}_{k+1}, k + 1)$ .

The above equations are used in a recursive *forward dynamic programming* procedure which works as follows (from [133] with elaboration):

(1) Quantize the state space  $[\mathbf{x}, t]$  to obtain a grid *consistent with the accuracy requirements of the problem*. The italicized words are quoted directly from Larson and Peschon, and speak to the potential pitfalls inherent in the discretization process. This implied warning is common to all implementations of dynamic programming for naturally continuous processes, as noted above.

- (2) Initialize the (forward DP) iterative procedure by defining  $I^*(\mathbf{x}_0, 0) = p(\mathbf{x}_0)$ , the *a priori* probability density for each possible discrete  $\mathbf{x}$  at time  $t_0$ .
- (3) For each quantized state  $\mathbf{x}_1$  (i.e., each possible discrete  $\mathbf{x}$  at time  $t_1$ ), calculate  $I^*(\mathbf{x}_1, 1)$  from  $\mathbf{z}_1$  and Eqn. (2.35), with appropriate subscript changes for stage 1, rather than  $k + 1$ .
- (4) Write  $\hat{\mathbf{x}}_0(\mathbf{x}_1, 1)$  as the value of  $\mathbf{x}_0$  for which Eqn. (2.35) is maximized in the previous calculation (establishing "pointers" which will be retraced to find the optimum state sequence).
- (5) Repeat steps (3) and (4) at each sampling instant until the  $k$ -th instant is reached. Each such iteration is one *stage*. This is the iterative forward dynamic programming procedure, moving forward through successive stages to completion.
- (6) Determine the *modal trajectory*  $\hat{\mathbf{X}}_{k/k}$  by first using Eqn. (2.31) to find  $\hat{\mathbf{x}}_{k/k}$  (i.e., the state with highest probability of being the terminus of the true state sequence) and then iteratively retracing the pointers set up in step(s) (4), to find the optimal state sequence, i.e.,  $\hat{\mathbf{x}}_{i/k} = \hat{\mathbf{x}}_i(\hat{\mathbf{x}}_{i+1/k}, i + 1)$ .

Thus, the Larson and Peschon equations define a forward dynamic programming algorithm for determining the one sequence of states  $\hat{\mathbf{X}}_{k/k}$  that gives us the maximum likelihood of generating a set of measurements  $\mathbf{Z}_k$ , given some additional or *a priori* information  $p(\mathbf{x}_0)$  and  $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$  about the likelihood of starting states and transitions in the state space.

**2.4.5 Relating Dynamic Time Warping (DTW) and the L&P Algorithm.** Both DTW and the L&P algorithm are DP sequence comparison techniques. The fundamental difference between them is that DTW does not consider state (as opposed to feature) transitions that occur off a single "one-dimensional" path in state space. In the usual DTW case, we have little knowledge of the underlying state space – only examples of the feature sequences produced by typical state trajectories. Observations from one state trajectory are simply compared to observations from another trajectory, and "warped" to allow for an optimal match. DTW generally attempts to associate an element of one sequence with more than one element of the other sequence, leading toward a bias for solutions that minimize the total number of associations.

On the other hand, the L&P algorithm can use information known *a priori*, or *aside* from the feature observations, about the likelihood of transitions in the state space. This allows the L&P algorithm to "investigate" more than one state trajectory. The L&P algorithm does not attempt to match more than one state space point with a given element in the feature sequence, and thus has no "arithmetic" bias toward short paths in the state space.

The drawback to the L&P algorithm is its "maximum likelihood" (ML) nature, in the sense that, given a set of  $m$  observations, it finds the set of  $m$  discrete states most likely to have generated the observations, subject to *a priori* constraints:  $p(\mathbf{x}_0)$  and  $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ . It may be, however, that a state space *region* exists which has a higher overall probability of producing the given observations, when all possible trajectories over time through that region are considered. By comparison with a region chosen by the L&P algorithm; this "better" region might have many points which are rather likely to have generated the given observations, while the "L&P" region has a few well-positioned points which are very likely origins, but many that are quite unlikely. The use of DTW in such a case, forcing each point along a likely state trajectory to associate with an observation, could select the "better" region instead of that selected by the L&P algorithm.

Unfortunately, while the L&P algorithm can use the (relative) computational economy of DP to find the ML sequence of states in a state space of arbitrary dimension, the state space region with highest probability of generating the observed features can in general be found only by exhaustive search. A set of nominal or *a priori* likely trajectories through the state space would provide a starting point for such a search with DTW methods. Clearly, a reasonable choice for the most likely trajectory based on *a priori* information is given by propagating the starting state and transition information used in the Larson and Peschon approach through the state space, without considering feature observations at each step. In other words, we start with  $\arg_{\mathbf{x}_0} [\max_{\mathbf{x}_0} p(\mathbf{x}_0)]$  and subsequently pick  $\arg_{\mathbf{x}_{k+1}} [\max_{\mathbf{x}_{k+1}} p(\mathbf{x}_{k+1} | \mathbf{x}_k)]$  over all  $t_k$  of interest. Chapters III and V will show the application of these concepts in object recognition applications.

Like all forms of classical sequence comparison, the L&P algorithm and DTW are fundamentally syntactic identification techniques. Implicitly, comparing two sequences

(or, as we will see in the following chapter, comparing a sequence with a space of possible sequences) using these techniques is a syntactic comparison process. We may not be able formally to identify the root symbol, production rules and other syntactic descriptors, but comparing the sequence terminals in proper order implies a comparison in some sense of the syntactic processes which generated those sequences. Interestingly, in the case of the L&P algorithm, we may note that the state transition likelihood quantity  $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$  and the measurement likelihood  $p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})$  for all states of interest are explicitly a *production rule* in the syntactic sense – for any one object class or grammar, they tell us how a sequence of states becomes a sequence of measurements [90:54] [212:318]. In cases of interest to us, the syntactic root symbols will be dynamic object classes exhibiting specific state transitions.

## 2.5 *Dynamic Programming in Object Recognition*

The purpose of this section is to illuminate previous efforts in object or target recognition and tracking that employed dynamic programming. Here and in the following chapters, these approaches will be contrasted with the author's research.

**2.5.1 *Conception and Development of Motion Warping.*** The fundamental inspiration for the research described in this dissertation was the observation that dynamic programming-based sequence comparison techniques could be applied meaningfully to object recognition using sequences of sensor signatures from a turning object, where (1) those signatures were a function of the angular orientation of the object, and (2) a close relationship exists between the object's translational and angular orientation states. The proposed concept is referred to as "motion warping". In summary, the key to this formulation is that we:

(1) Start with a potential object's underlying "feature observable surface," nominally a hypothetical aspect angle sphere coordinatized in some two-dimensional (azimuth and elevation) spherical coordinate representation (but, like the earth's surface, suitable for analysis as a planar section over local regions), with each (angular) dimension coordinatized linearly.

(2) "Pre-warp" a one or two dimensional subset of that surface to make it appear as it would to a sensor scanning it at locations and rates estimated from kinematics and object dynamic restrictions.

(3) Compare the observed measurement sequence with the pre-warped surface subset, treating the comparison as one of finding the most likely continuous path for the observed sequence along the pre-warped surface, allowing for additional warping by dynamic programming-based sequence comparison techniques (e.g., dynamic time warping, not to be confused with the *pre-warping* process).

(4) Select, as the most likely object class corresponding to the observed measurements, the potential object class offering the closest match between the observed sequence and its pre-warped surface subset.

With this basic plan in mind, the author reviewed relevant sources on multisensor fusion, object tracking, and pattern recognition to determine if the proposed concept had already been investigated. Eventually, this research included all of the classified resources available from the Target Recognition Technology Branch (WL/AARA [166]) of Wright Laboratory at Wright Patterson Air Force Base, Ohio.

The first three of the sources discussed below (Kenyon, Gorman, and Amini *et al.*) were among the first to be found and reviewed, and were readily eliminated from consideration as competing research. The subsequent three sources (Larson and Peschon, Barniv, and Kramer) served as inspiration for recent efforts by the author of the final reference (Mieras *et al.*), whose methods approach but do not encompass the research described in this dissertation.

Between Kramer and Mieras, however, we consider a much older (1978) reference by Le Chevalier *et al.* of France whose concepts are very close to those of Mieras *et al.* and similarly approached (*particularly in philosophy*), but did not encompass, the author's concepts. These two final authors, Le Chevalier and Mieras, are the only sources found whose concepts resemble motion warping to any meaningful degree, so their efforts are of particular interest.

In the next chapter, we will see that the efforts of Le Chevalier and Mieras can be posed as suboptimal versions of the methodology pursued in this research. This research serves to generalize and significantly extend their groundbreaking efforts.

*2.5.2 Previous Developments by S. Kenyon.* In a 1988 article [122] in the Proceedings of the 1st National Symposium on Sensor Fusion [42], Kenyon proposed to use dynamic programming with object kinematics and attributes in an objective function to match observations with track files. Thus, the intended use of the proposed system was as for the systems proposed by Bar-Shalom, Blackman, and Mitzel in Sect. 2.3.3 (see references there). Since the Kenyon article was not specific as to how dynamic programming was employed, Kenyon was contacted telephonically [123] and was asked, after a short explanation of the proposed research, if his article referred to a concept like motion warping. He replied that it did not.

*2.5.3 Previous Developments by A. Amini, H. Yamada, et al.* In the course of researching previous efforts to apply dynamic programming in pattern recognition, a number of articles were found regarding efforts to match *stationary* images of objects to *a priori*-defined images or image maps. Representative of these are [70, 3, 230]. These techniques are primarily point-to-point correspondence techniques as discussed in App. B, with dynamic programming used to establish a minimum-cost association between points. This effort has no similarity with the author's research.

*2.5.4 Previous Developments by B. Burg, J. Gorman, et al.* A 1986 article by Burg and Zavidovique [41] of France proposed the use of dynamic programming-based sequence comparison mathematics of the type proposed for this research to recognize *stationary* images. A 1988 article by Gorman [101] proposed the similar use of dynamic programming-based sequence comparison techniques to recognize partially occluded stationary objects. Basically, Gorman's research called for breaking up an observed candidate object silhouette (possibly partially occluded) into a set of sections, for each of which a Fourier descriptor representation (analogous to a *word*) was defined. These section representations were then compared to the sections for full silhouettes from known object



classes and orientations. The object was to find that full silhouette which best matched the observed silhouette over most of its perimeter, with non-matching portions ascribed to partial occlusion. Due to the similarity of this effort with the proposed effort in terms of application and mathematical tool, Gorman was telephonically contacted [100] and asked if his research had included anything like motion warping. He stated that it had not – in particular, he had not considered the issue of changing aspect angle. So far as can be determined, also, the work of Burg and Zavidovique has no impact on the author's research.

*2.5.5 Previous Developments by Y. Barniv et al.* The developments by Y. Barniv, as originally published in [12, 13] for low signal-to-noise ratio object tracking using dynamic programming, are a direct application and development of the Larson and Peschon equations (Sect. 2.4.4). Together with later ideas advanced by Weiss and Friedlander [222], Barniv's concepts appear in their latest form in [8:85–154] (note: the reader is advised to read Larson and Peschon [133] before reading the Barniv works). The apparent intended application of Barniv's concepts, or at least *an* application which serves to illustrate the concept, was as a signal processor for a space-based imaging infrared sensor attempting to detect cruise missiles over the ocean. The Barniv procedure is one of a class of what are called "*track before detect*" algorithms, i.e., algorithms that use some rules to string together sequences of states or points in space as though they constitute tracks, and *then* attempt to determine if an object was actually passing through any of the sequences of points at the observed times.

Barniv's contribution was to define the terms in the Larson and Peschon Eqns. (2.30) through (2.35) appropriately for his chosen problem. In the Barniv development, one tracks motion on a  $32 \times 32$  array of imaging sensor pixels. Each pixel is divided into 4 quadrants, called cells. Thus the image plane has  $64 \times 64$  cells. A "*state*" is a pairing of any two cells, a start cell and an end cell, located 4 pixel or 8 cell lengths apart. This distance corresponds to the travel of a nominal object over the pixel integration time. For any given cell, there are 164 states corresponding to trajectories in all possible directions. Thus, on

the image plane, there are  $64 \times 64 \times 164 = 671,744$  total states. A stage in the dynamic programming sense is defined by all possible states over the pixel integration time.

The discrete probability density  $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$  is defined through any one of several approaches as a monotonically decreasing function of the absolute value of the curvature of the trajectory – that is, reflecting the fact that cruise missiles are most likely to travel in straight lines. Since a given state or straight line trajectory segment passes over given rectangular cells with generally different dwell times in each, this factor is taken into account when defining  $p(\mathbf{z}_k | \mathbf{x}_k)$  – the probability or likelihood of obtaining the observed measurement from a set of cells, given that the trajectory passed through that set of cells. Basically, then, the Larson and Peschon procedure is repeated, with each new set of integrated image frames, starting with every cell/direction combination on the image plane as a possible  $\mathbf{x}_0$ , looking for generally straight tracks over a growing number of image frames, and identifying the latest endpoints of potential tracks from states (effectively, terminal cells with some inbound direction) that have a value of  $I^*(\mathbf{x}_k, k)$  greater than some threshold.

*2.5.6 Previous Developments by Kramer and Demirbağ.* The concepts proposed by Barniv were subsequently adapted by Kramer and Reid [129] for “*track before detect*” processing with doppler radar. In the Kramer development, the states are “*range-azimuth-doppler*” cells, i.e., discretized range and azimuth cells that are further subdivided according to doppler velocity bins (elevation is not considered explicitly, so the physical tracking space is apparently two-dimensional). Thus, Kramer starts with each range-azimuth-doppler cell, performing a forward dynamic programming procedure after each scan to determine, for each cell with some minimum acceptable return level, what the most likely predecessor cell was. The analog of Larson and Peschon’s  $I^*(\mathbf{x}_k, k)$  parameter here is Kramer’s track “*quality count*,” which is not a probability, but which serves to identify likely object tracks.

Demirbağ [66] has (apparently independently) developed a similar concept for radar tracking. The author has not analyzed his concepts in detail.

2.5.7 *Previous Developments by F. Le Chevalier et al.* Several months after the definition and development of the motion warping concept, which included extensive research covering the time period from 1980 on, and which indicated no similar efforts in radar or imaging sensor object recognition or computer vision [14, 10, 8, 28, 33, 40, 55, 94, 96, 123, 166, 169, 180, 191, 218, 224], the author conducted additional research into the pre-1980 time frame on a related concept. While reviewing the *Proceedings of the 1978 IEEE Conference on Pattern Recognition*, the author found an article by François Le Chevalier of O.N.E.R.A., Paris, France [136] which included several of the same observations and ideas as those behind the author's concept of motion warping. Specifically, the key points of the Le Chevalier article were that:

- (1) Aspect angle information for aircraft targets would be available from a radar tracker and could be used to fuse radar cross section (RCS) measurements for improved identification.
- (2) A "shortest path" (presumably, Le Chevalier meant forward dynamic programming) process linking associations between (a) model outputs at various aspect angles and (b) measurements from a target of unknown type could be used to determine the most likely target class to have produced the observed sequence of outputs.
- (3) The transition rules in the dynamic programming process would be driven by "evolutionary constraints" associated with the target dynamics as observed by the radar – i.e., model aspect angle state transitions which did not conform to feasible or observed target dynamics would not be allowed. These constraints are evidently invoked as limits, or "yes/no" decisions, and, as will be seen in the following chapter, are therefore invoked more crudely than the approach envisioned in the author's research.
- (4) The proposed method could be employed with sensor systems other than radar – indeed, for any feature observable for which a metric between measurements and predictions can be established.

This proposal by Le Chevalier *et al.* was derived not from an understanding of speech processing and multisensor fusion as was the author's, but from an appreciation that a radar target could be represented as a finite state automaton producing sequences of observables over time/aspect angle, in accordance with evolutionary constraints described

by the target's dynamic limitations. Undeniably, the motivation and goal of Le Chevalier *et al.* were the same, and the methods are related, as we will see below. The research described herein, however, was developed independently of Le Chevalier's work.

Reviewing the discussion in [136], the Le Chevalier technique is believed to work essentially in the following fashion. First of all, from some (presumably kinematic) "*a priori*" aspect angle information, a given (first) radar signature is known to originate from some relatively large aspect angle window on the unknown target, say 10 to 20 degrees square. The aspect angles in this window are defined for discrete values of length and width (linear aspect angle). Then, the feature space metric matching distances between the observed signature and the model-defined signatures are defined for each discrete aspect angle value in the window, for each model.

Due to the Chi-square metric used by Le Chevalier *et al.*, this matching distance, for any given observation-to-model/aspect angle comparison, is directly related to the maximum classical likelihood (probability density value)  $p(\mathbf{z}_k^f \mid \omega_i, \mathbf{x}_k^a)$  that feature observation  $\mathbf{z}_k^f$  was generated by model  $\omega_i$  at discrete aspect angle state  $\mathbf{x}_k^a$ .

Now, we consider the second observed signature, also defined to have come from some nominal aspect angle window on the target, possibly a somewhat different window than on the last signature. Again, for each model and window (at suitably discretized aspect angles) we define the Chi-square (matching) distances to this second observed signature. Next, for each model, around each discrete aspect angle value (state) with a matching distance less than some threshold for the *second* signature, we examine the matching distance values for the *first* signature from states that lie within some "evolutionary constraint" aspect angle bound or "association gate" (evidently circular) around that second signature/aspect angle match. We associate each second signature/aspect angle match with the most likely (minimum matching distance) first signature/aspect angle match lying within the acceptable radius. The total matching distance associated with the second state is now the sum of the second and (minimum) first state distances. This process continues over successive signatures for each model until the sequence with minimum total matching distance to the final window (signature) is chosen, indicating the most likely model to have produced the observed measurement sequence, and a corresponding aspect angle history.

Significantly for our later discussion, Le Chevalier appears to forego explicitly the opportunity to "propagate" the target angular state according to observed kinematics. He maintains that restricting allowable transitions to known "evolutionary constraints" will reveal the correct target class and aspect angle path without the need for detailed knowledge of target kinematics. As we will see in the following chapter, Le Chevalier's claim that his approach works "in real time" for aircraft targets is further indication that he eschews the propagation of angular states according to observed kinematics.

Subsequently, the author conducted detailed research to determine the extent to which Le Chevalier *et al.* had developed the concept outlined in the 1978 article. Further articles by Le Chevalier and references to his work included the following (no further works by his co-authors were found):

- (1) A classified article published in French as part of the 1980 AGARD Conference on "Image and Sensor Data Processing for Target Acquisition and Recognition," which resulted in AGARD Conference Proceedings CP-290 of the same title [137].
- (2) A French patent, number 2402971, awarded in April 1979 [135], the application (number 7727362) for which was referenced in the 1978 article. This patent covers the ideas expressed in the 1978 article.
- (3) A reference to Le Chevalier's 1978 article as an application of pattern recognition for target identification, without further comment, in a 1980 overview on (then) recent advances in pattern recognition by the well-known pattern recognition expert, the late Dr. K.S. Fu [89].
- (4) A paper by Le Chevalier delivered at, and published in the proceedings of, the 1984 International Conference on Radar (not IEEE-sponsored) in Paris [138].
- (5) A short discussion of Le Chevalier's 1978 article in a generally excellent encyclopedic Russian work [172] covering all forms of radar target recognition known to have been published in the West. As an aside, the referenced text is available in English from the Defense Technical Information Center (DTIC), and is highly recommended for any target recognition researcher.

(6) An article by Le Chevalier published in the 1986 edition of *La Recherche Aerospatiale* (the organizational journal of O.N.E.R.A., where Le Chevalier was employed) [134].

(7) A reference to Le Chevalier's 1978 and 1984 works in a chapter on target recognition considerations in a 1987 book entitled *Principles of Modern Radar* [76]. The chapter was written by Dr. N.F. Ezquerro, now (1991) employed by Georgia Tech Research Institute in Atlanta, GA.

(8) A reference to Le Chevalier's 1986 work in a 1990 *Recherche Aerospatiale* article by another O.N.E.R.A. employee [25].

(9) A U.S. patent, number 4,735,379, awarded 5 April 1988 [142], which refers to the earlier French patent number 2402971.

The author initially obtained the works listed as (3), (4), (6), and (7) above, and telephoned Dr. Ezquerro (see item (7)) to question him regarding the state of development of Le Chevalier's concept. Neither item (4) nor item (6) discuss the subject of the 1978 article at all. Dr. Ezquerro stated [80] that he had co-chaired a panel with Le Chevalier at the 1984 Conference, and that, so far as he (Ezquerro) knew, no further development of Le Chevalier's work has been made.

Ultimately, the author obtained the other items listed. Only items (1), (2), and (9) mention further applications of the 1978 proposal. Clearly, the patents, or items (2) and (9), are of key interest. Item (2) does not amplify the 1978 proposal in any significant way as far as this research is concerned. In an extension of that article, however, item (2) expresses the intent to use this approach in an application similar to that of Kramer and Demirbaş, as discussed in the preceding Sect. 2.5.6. Item (9) applies the 1978 concept to searches on the earth's surface, rather than in aspect angle on a target signature library. Again, this is functionally equivalent to the Kramer concept, although aspects of the Kramer concept appear to be improvements over the Le Chevalier approach.

Item (1) refers briefly to the applicability of Le Chevalier's concept to HRR radar target recognition, but does not develop the theory of the concept further. The fact that this source is printed only in French in a classified, relatively obscure (at least, very unfortunately, for Americans) journal may explain the fact that the author has seen no

references to it anywhere. Item (5) essentially repeats the 1978 Le Chevalier article, but misses the point about the importance of dynamic programming to the method.

Noting Le Chevalier's claim in the 1978 article that his work was an original contribution to the study of syntactic methods of pattern recognition as discussed by Fu in [88] and other works, and Fu's brief mention of the Le Chevalier article in Item (3) above [89], the author reviewed subsequent books on syntactic/structural pattern recognition by Fu [90] (pub. 1982) and Miclet [161] (pub. in French, 1984; in English, 1986). Neither of these extensive works contain any mention whatsoever of Le Chevalier's efforts. In addition, the author reviewed the references for all books and articles reviewed previously (including all other references in the bibliography of this work). No other references to the Le Chevalier work were found.

One other reference *was* found, however, that closely follows the Le Chevalier approach, but derives from a different origin. That effort is the subject of the following section.

*2.5.8 Recent Developments by H. Mieras et al.* In reviewing the Proceedings of the 1990 Combat Identification Systems Conference [231], the author found an article [164] by H. Mieras *et al.* of Raytheon in which the authors claimed to have made use of dynamic programming for "integrating" high resolution radar range sweeps over stable or changing target aspect angles. Mr. Mieras was telephonically contacted [162] and questioned about his efforts. He stated that their approach (considered proprietary by Raytheon) was inspired by the Kramer effort [129] discussed above, but working with aspect angle "bins" (discretized values) rather than radar range/angle/doppler bins. Subsequently, a more detailed report on the radar target recognition effort discussed in [164] was found to be available in [165]. The Mieras effort is independent of, and evidently somewhat extends, the effort of Le Chevalier.

Like the Le Chevalier approach, the Mieras approach uses forward dynamic programming to establish a minimum distance match between a set of signature observations and library signatures stored as a function of aspect angle for any candidate target class. Due to the Mahalanobis metric and the range alignment technique used by Mieras *et al.* (see

Sect. 2.2.3), this distance, for any given observation-to-model/aspect angle comparison, is directly related to the maximum classical likelihood (probability density)  $p(\mathbf{z}_k^f \mid \omega_i, \mathbf{x}_k^a)$  that observation  $\mathbf{z}_k^f$  was generated by model  $\omega_i$  at discrete aspect angle state  $\mathbf{x}_k^a$ .

In addition to Kramer [129], Mieras also noted as references the articles by Barniv [12, 13] noted earlier, and another by Scharf and Elliot [196] (the latter is a relatively broad 1981 overview on dynamic programming in signal and image processing). Mieras was not aware of the work by Le Chevalier [136], and did not explicitly mention that of Larson and Peschon [133].

Of particular interest to the author was the way in which the Mieras method uses *a priori* aspect angle data, derived for example as we have discussed in Sect. 2.3.3.1 from kinematics. Mieras noted that his aspect angle information was limited to "a rough estimate, say within a 10 or 20 degree window at the time of the sweep" [162]. He also noted that his algorithm did not establish an aspect angle path before the processing, but that one "comes out of the algorithm" [162]. In response to further questions, he noted that constraints on sweep to sweep continuity were "set" at a particular angular value. More recent communication with Mieras [163] appears to indicate that his algorithm biases the angular constraints, or a circular "association gate" like that discussed above for the Le Chevalier approach, in the direction expected according to observed kinematics, but does not use formal aspect angle transition likelihoods, which we will see are available from the tracking process.

These comments by Mieras and the referenced articles indicated to the author that the Mieras method does not fully exploit or "fuse" information available from kinematics, which is the fundamental intent of this research. To the level of description available, Mieras' technique apparently corresponds closely to procedures described for the Le Chevalier method – matching  $n$  observations to  $n$  locations in an aspect angle state space, subject to kinematic transition limitations. The steps taken by Mieras to bias angular constraints in the direction expected according to observed kinematics and other techniques [162] appear to make Mieras method an improvement over that of Le Chevalier. The differences between these methods and the accomplished research of the author will be highlighted in the following chapter.



*2.5.9 Significance of the Le Chevalier and Mieras Efforts.* The core inspiration behind the research described in this dissertation was the observation that dynamic programming sequence comparison could be a key tool for multisensor fusion and dynamic object recognition. Clearly, Le Chevalier and Mieras have independently preceded the author in the kernel of that observation. The author's real contribution, then, is to develop further the theory and methods for use of dynamic programming and other sequence comparison methods for these purposes. The techniques of Le Chevalier and Mieras will be shown to be elements in a family of more-or-less capable algorithms for achieving a particular purpose in multisensor fusion – exploiting the joint likelihood of all observable events. The author strongly believes that the class of approaches conceived by Le Chevalier, Mieras, and the author has not heretofore received the development or application that it deserves.

## 2.6 Multisensor Fusion

**2.6.1 General Overview.** Multisensor fusion is an extremely broad concept which encompasses all techniques for melding information from one or more sensors to make better (i.e., more probably successful) decisions than could be made using the output of individual sensors separately. Taxonomies for multisensor fusion are more diverse than those for pattern recognition, and this work will not delineate any – the reader is referred to any of [42, 43, 44, 106, 107, 126, 145, 146, 189, 218, 224]. From [218:1], we have the following definition for *data* fusion (equivalent to *multisensor* fusion from our perspective), in a military context:

(multisensor or data fusion is) a multilevel, multifaceted process dealing with the detection, association, correlation, estimation, and combination of data and information from multiple sources to achieve refined state and identity estimation, and complete and timely assessments of situation and threat.

It should be recognized by the reader that *all* of the estimation and recognition approaches discussed thus far in this chapter are sensor fusion techniques – even the lowly  $\alpha$ - $\beta$  tracker, fusing information *from one sensor over time*, can be considered a sensor fusion device. In Sect. 2.2, we discussed fusion of feature observable information to define the pattern class and orientation for an unknown object. In Sect. 2.3.2, we discussed fusion of kinematic information to determine object track. Finally, in Sect. 2.3.3, we discussed fusion of kinematic *and* feature observable information to determine object tracks.

Perhaps the one fundamental principle behind more forms of multisensor fusion than any other is Bayes' Rule [197, 177], underlying as noted above a major branch of pattern recognition as well as neural net theory [190] and the Kalman filter [153:209–217]. As we saw in Sect. 2.2.1, however, the full potential of Bayes' rule can only be reached where we know all possible classes of events (in our case, objects and their orientations), the *a priori* probability of the occurrence of those events, and the probability (likelihood) that a given actual event generates a given observation.

Lacking this *a priori* information, the principal alternative is *maximum likelihood* estimation or classification (e.g., Eqn. (2.2)). For example, if we have no *a priori* information on object or orientation probability, but know all possible classes of objects

and have the classical likelihood function  $p(\mathbf{z} \mid \omega_i)$  that each candidate object  $\omega_i$  generates any feasible observation  $\mathbf{z}$ , we may assign the unclassified object to the class which maximizes that classical likelihood function for the actual observed  $\mathbf{z}$ . Maximum likelihood classification can be used suboptimally even where we do not know all possible classes of objects, by assigning an unclassified object to the *known* class which maximizes the classical likelihood function. Maximum likelihood estimation is identified by some [10, 33, 36, 82, 106] [224:6] [218:215] to be a distinctively important technique for multisensor fusion, although other authors [145, 146] do not explicitly discuss it, presumably lumping it together with other probabilistic techniques. It is important to understand, however, that the output of a maximum likelihood estimator need not be a probability measure *per se*, as we will discuss in Sect. 2.7.

Considering the use of maximum likelihood classification when all classes are *not* known, the reader may observe that there is an unquantified probability that the observed measurements do not correspond to any of the known objects. For cases like this, where some lack of information introduces the need to quantify uncertainty, the Dempster-Shafer technique is available [8, 33, 218]. The Dempster-Shafer methodology has been called a generalization of Bayes' Rule to allow for uncertainty [33:381–386]. The Dempster-Shafer analog to probability is probability mass, representing knowledge. This mass (totalling to a value of one) may be allocated in combinations of four different ways: (1) to *support* (confirmation) of any one particular hypothesis (say  $\mathcal{H}_1$ ); (2) to *uncertainty* (say  $\mathcal{U}$ ), confirming no hypothesis; (3) to a disjunction of hypotheses – say  $\mathcal{H}_1 \vee \mathcal{H}_2 \vee \dots \vee \mathcal{H}_n$ , where  $\vee$  (*or*) indicates logical disjunction (i.e., information that indicates that  $\mathcal{H}_1$ , *or* some other hypothesis, is true); or finally (4) to *negation* of any particular hypothesis  $\mathcal{H}_1$ , confirming for example that  $\mathcal{H}_1$  is *not* true. All mass not allocated to the negation of a particular hypothesis is taken to indicate the *plausibility* of that hypothesis.

Just as with a Bayesian classifier, where the addition of measured information ideally causes the *a posteriori* probability of one hypothesis to rise above the others, the addition of information to a Dempster-Shafer classifier causes probability mass to be reallocated, ideally causing a preponderance of mass to be assigned to the *support* category for one

of the known hypotheses. When hypotheses do not overlap, and there is no uncertainty, Bayes' Rule and Dempster-Shafer logic produce equivalent results.

*2.6.2 Multisensor Fusion for Dynamic Objects.* In this section, we will address the question of why a multisensor fusion algorithm which fuses feature observable and kinematic information should be able to recognize or discriminate objects better than an algorithm using either type of information alone. The fundamental point to be made here is that by increasing the number of test conditions that an unknown object class must "pass" in order to be identified as a member of a known class, we reduce the probability that an unclassified object could pass the tests for an *incorrect* class. Said another way, adding kinematic or other requirements reduces the *joint likelihood* that the unclassified object could exhibit the particular combination of behavior corresponding to an incorrect class.

These observations regarding the significance of joint likelihood for improved multisensor fusion and object recognition are not original to this effort. The unique aspects of this research are that we provide (1) new understanding as to why it is important to consider the joint likelihood of kinematic and "nonkinematic" or feature observable events, and (2) a class of tools – sequence comparison algorithms and multiple model estimators (e.g., composed of Kalman filters) – for exploiting the joint likelihood of observed events *over time* in dynamic object or target recognition.

Most automatic object recognition algorithms use feature observable information only – those that are Bayesian in nature, providing an *a posteriori* probability of class membership as in Eqn. (2.1), can be characterized as providing the *a posteriori* probability  $p(\omega_i | \mathbf{Z}_k^f)$  that we are actually observing an object of class  $\omega_i$ , given a set of  $k$  feature or signature measurements  $\mathbf{Z}_k^f = \{z_1^f, z_2^f, z_3^f, \dots, z_k^f\}$  (where the subscripts imply measurement respectively at discrete times  $t_1$  through  $t_k$ , and the superscript stands for *feature*), and *a priori* object class probabilities  $p(\omega_i)$  for each of  $J$  known object classes. Classically, the measurements are treated as independent in time, or from measurement to measurement.

Using a Bayesian approach *and* fusing kinematic information, however, we would really desire to produce a pattern recognition system that estimates the *a posteriori* probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  that we are actually observing an object of class  $\omega_i$ , given a set of  $k$  feature or signature measurements as above,  $m$  kinematic measurements  $\mathbf{Z}_m^d = \{\mathbf{z}_1^d, \mathbf{z}_2^d, \mathbf{z}_3^d, \dots, \mathbf{z}_m^d\}$  (note use of superscript  $d$ , as in “dynamic”, rather than  $k$  as in “kinematic”, since  $k$  is a counting index in the Larson and Peschon form), and *a priori* object class probabilities  $p(\omega_i)$ . The “kinematic” measurements correspond classically to positions and velocities at discrete times  $t_1$  through  $t_m$  – the likely relationship between  $t_1$  through  $t_k$  and  $t_1$  through  $t_m$  is discussed in Sect. 3.6.5.

This ideal, but in practice generally unobtainable, system would consist of  $J$  functions, one for each object class, having a domain of the space of all measurements over time and a range on the interval of the real line from zero to one, with the sum of the  $J$  function values equal to one (or less than one, if we wish to allow for unknown classes). Following Rao [184:353], however, in the absence of  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  (or equivalently, the joint probability density  $p(\omega_i, \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  of object class and measurements), we must be content to find a set of “generalized” likelihood functions such that the maximum value for each function is attained for the correct combination of object class and measurements. This likelihood function may not provide a probability measure *per se*.

More specifically, a generalized likelihood function  $L[\mathbf{x}, \mathbf{y}, \mathbf{Z}_k]$  [154:75] is simply a real-valued function of states  $\mathbf{x}$ , parameters  $\mathbf{y}$ , and *measurement history*  $\mathbf{Z}_k$  (i.e., measurements  $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \dots$ , up through and including some latest measurement  $\mathbf{z}_k$  at time  $t_k$ ), which is defined *a priori* for a given set of states  $\mathbf{x}$  (a maneuver) and parameters  $\mathbf{y}$  (an object). The distinction between states and parameters made here is again that of Maybeck [154:69], as in Sect. 2.3.1.3. We require only that this function consistently achieve a maximum value for measurements  $\mathbf{Z}_k$  taken from a *truth model* or actual system having the same values of (object) parameters and states – in other words, the likelihood function is a “matched filter” which should exhibit high “gain” only for signals with its design specifications. Clearly, the object signature libraries, aspect angle “windows”, and associated metrics used in classical automatic object recognition (e.g., [20]) are likelihood functions.

What can we gain by considering the kinematics of the unknown object? Consider an abstract space  $\Theta$  of all possible object models and aspect angles over time as part of the domain of a matching function,  $\mathcal{L}$ . The remainder of the matching function domain lies in another abstract space  $\mathbf{Z}^f$  of noise-corrupted feature observations. The matching function operates on an element from  $\Theta$  and an element from  $\mathbf{Z}^f$  to produce a scalar value, the magnitude of which is some measure of the likelihood that the chosen element in  $\Theta$  could produce the observed element in  $\mathbf{Z}^f$ .

A classical generalized likelihood function for recognition of a particular object class  $\omega_i$  is defined by restricting the domain of  $\mathcal{L}$  to produce an  $\mathcal{L}_i$  with domain  $\mathbf{S}_i \subseteq \Theta$  corresponding to  $\omega_i$ . Typically, we match sets of noise-corrupted feature observations from abstract space  $\mathbf{Z}^f$  over time to elements in the first space. Unfortunately, these classical *decision theoretic* functions may give higher likelihoods than ideal for the *wrong* object class, in part because kinematically unlikely aspect angles (pose estimates) and aspect angle transitions over time are allowed. Conventional algorithms that limit searches in aspect angle to "windows" defined by kinematics are a step in the right direction, but these algorithms are still "independent look" algorithms as defined in Sect. 2.2.1, and wild pose estimate transitions may occur from measurement to measurement.

The key to the author's approach is to restrict the domain of each matching function  $\mathcal{L}_i$  further, requiring the object aspect angle over time to be consistent with the observed kinematics *and vice versa*, since this restriction (correctly executed) should not adversely affect matching or likelihood function values for measurements from the *correct* object class, but may lower the values given by matching functions corresponding to *incorrect* object classes. This approach for understanding the need to fuse "kinematic" and "nonkinematic" or feature observable information is an original contribution of this effort. Despite the general lack of a full expression for the joint classical likelihood  $p(\omega_i, \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  of object class and measurements, we still seek to exploit the *joint* nature of observable events by finding the object class most likely to have exhibited simultaneously the behavior observed over time in several domains (kinematic, feature observable, etc.).

As we will show, prior efforts have moved in this direction by restricting the matching or likelihood function domain to be consistent with *feasible* kinematics [136], and, in a

suboptimal fashion, consistent with *observed* kinematics [164, 165, 163]. By further, *optimal* restriction using observed kinematics, we will achieve a more highly “tuned” likelihood function (again using the matched filter analogy). Simply, restricting the matching domain of a likelihood function according to kinematics is the analog of conditioning  $p(\omega_i | \mathbf{Z}_k')$ , were it known, on the added information given by kinematic measurements  $\mathbf{Z}_m^d$ .

The preceding discussion correctly implies that we will always have equal or better recognition performance in identifying a particular object in a particular maneuver if we progressively restrict the domain (in time, space, frequency, or any other dimension) of each candidate matching or likelihood function to subsets that exhibit only the correct combination of behavior for the corresponding object – and no more. The correct object-to-likelihood function match will still indicate high likelihoods – likelihoods for incorrect matches are more likely to fall.

Mathematically, the concept of restricting the matching domain for equal or better performance can be expressed as shown in Eqn. (2.36) below – for any likelihood function  $\mathcal{L}_i$  corresponding to object class  $\omega_i$ , with the kinematically unrestricted and restricted matching domains denoted respectively by  $\mathbf{S}_i$  and  $\mathbf{T}_i$ , we can show trivially by contradiction that:

for  $\mathbf{T}_i \subseteq \mathbf{S}_i$ :

$$\sup \mathcal{L}_i(\mathbf{T}_i, \mathbf{Z}') \leq \sup \mathcal{L}_i(\mathbf{S}_i, \mathbf{Z}') \quad (2.36)$$

The preceding equation does not imply that *any* matching domain restriction will improve recognizer performance. Domain restriction must be done correctly – no likelihood function for an object class should fail to include areas in the domain that may contain feasible sets of behavior for that class. Since theoretically optimal domain restrictions may never be known, we expect that some *tuning* process may be required to define empirically optimal restrictions.

Using the same approach from another perspective, we may force kinematics to be consistent with aspect angle estimated from feature observations. Where an incorrect

unknown object-to-library match forces a tracker/filter dynamic model to deviate strongly from the true dynamics, this mismatch will be immediately evident through poor tracking performance, large filter residuals, and other factors. Note that, in effect, we have extended the abstract space  $\Theta$  to include not only object models and aspect angles, but kinematic (translational states and derivatives, and derivatives of the fundamental rotation states associated with aspect angles) and time dimensions as well – and our matching process now requires consistency across all dimensions, i.e., high *joint likelihood*.

This joint likelihood of kinematics and feature observables can be exploited in a Bayesian structure, naturally expressed in practice as a generalized likelihood function. This is the subject of the next chapter.

Finally, recalling the discussion of syntactic pattern recognition in Sect. 2.2.2, note that the preceding discussion is implicitly an argument for the application of syntactic methods in object recognition. In effect, we have said not only that particular *pattern primitives* are characteristic of particular objects, but their order of presentation, and the *productions* which govern transitions, are as well. The extremely powerful observation that syntactic techniques apply in target recognition was first made by Le Chevalier [136]. Another observation by Le Chevalier in the same source was that syntactic dynamic programming methods would reduce *ambiguity* in target recognition.

This last observation, independently made by the author and focused further in this research, led to the choice of generalized ambiguity functions for evaluating the performance of the proposed “generalized” likelihood function algorithms. The ease of making an analytical prediction of performance for a generalized likelihood function depends on the nature of the function. *Any* such function, however, can be evaluated in experiment or simulation with the use of a *generalized ambiguity function* (as proposed by Schweppe [198, 154]), which is the subject of Sect. 2.7 below.

## 2.7 Generalized Ambiguity Functions

This section follows the development given by Maybeck [154:96–101] to motivate the use of generalized ambiguity functions (GAF) for assessing maximum likelihood estimator



performance, as developed by Schweppe [198:376-381]. The generalized ambiguity function is an extension of the classical radar ambiguity function [57, 16] used to define radar pulse parameters for optimum information quality (balancing range resolution vs. doppler error, for example) according to system requirements.

The result of the research described in this dissertation is an object recognition approach which seeks to reduce ambiguity in recognition decisions by *fusing* all available information over short periods of time. The generalized ambiguity function discussed in this section will be shown to be a natural tool for evaluating the performance of this research product and, in general, any object recognition schemes that can be viewed as maximum likelihood estimators.

The generalized ambiguity function is defined by the following equation:

$$A_k(\Omega, \Omega_t) \equiv \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} L[\Omega, Z_k] f_{Z(t_k)|\Omega(t_k)}(Z_k | \Omega_t) dZ_k \quad (2.37)$$

where:

$\Omega_t$  = the particular combination of states  $\mathbf{x}(t)$  and parameters  $\mathbf{y}$  (generally constant over the time interval of interest) for the *truth* system which generates the set of all possible measurement histories  $Z_k$  over which the integral is taken – a likelihood function  $L$  defined for  $\Omega_t$ , operating on an element of this measurement history set, will ideally generate a higher value than will any  $L$  defined for some *other* value of  $\Omega$ , operating on an element of this measurement history set (the ambiguity function evaluates the extent to which this is true in the mean)

$\Omega$  = state/parameter values for which the generalized likelihood function is defined, for evaluation against measurements generated by a truth model with state and parameter values  $\Omega_t$

$A_k(\Omega, \Omega_t)$  = the generalized ambiguity function, a function of  $\Omega$  for a given  $\Omega_t$  and likelihood function  $L$

$L[\Omega, Z_k]$  = the generalized likelihood function, a function of  $Z_k$  when defined for a given  $\Omega$  (note that the calligraphic letter " $Z$ ," or  $Z$ , is used as the "dummy" form of  $\mathbf{Z}$ , appropriate for showing functional relationships in an integrand)

$f_{Z(t_k)|\Omega(t_k)}(Z_k | \Omega_t)$  = the probability density function of the measurements, given that the true states and parameters have the value  $\Omega_t$

$Z_k$  = the measurement history vector as of time  $t_k$ , as defined in Sect. 2.6

The reader must not allow the complexity of Eqn. (2.37) to obscure the essential simplicity of the concept. The generalized ambiguity function is the expected (or mean) value of a generalized likelihood function defined for varied combinations of states and parameters, conditioned on the true states and parameters having particular values. For any particular value of  $\Omega$  defining the likelihood function, there is in fact a distribution of likelihood function values produced, due to the different realizations of measurements produced by a system with true states and parameters  $\Omega_t$ .

Examining the behavior of the ambiguity function for each realizable value of  $\Omega_t$  and, for each such value of  $\Omega_t$ , a range of  $\Omega$  encompassing reasonable values of states and parameters expected other than at  $\Omega_t$ , we desire that the ambiguity function have an easily discernible global maximum at  $\Omega_t$  – i.e., that local maxima, if present, are "widely" separated from the global maximum at  $\Omega_t$ . Fig. 2.10 shows examples of "better" (than good), "good," "mediocre," and "poor" ambiguity functions, corresponding to different likelihood functions of respective quality defined over the same domain of state and parameter values.

It is important to understand the intent behind the grading terms applied in Fig. 2.10. The "good" ambiguity function is so rated because it has a single peak value, peaking at the value of  $\Omega = \Omega_t$  as desired, with the peak "rolling off" reasonably quickly as  $\Omega$  moves away from  $\Omega_t$ . The "better" ambiguity function is so rated because it has all of the qualities of the "good" function, but "rolls off" even more quickly (more precisely, the second derivative or curvature of the "better" function at the peak value is more negative). This means that an identification of true state and parameter values at  $\Omega_t$  can be made with less ambiguity than with the "good" function.

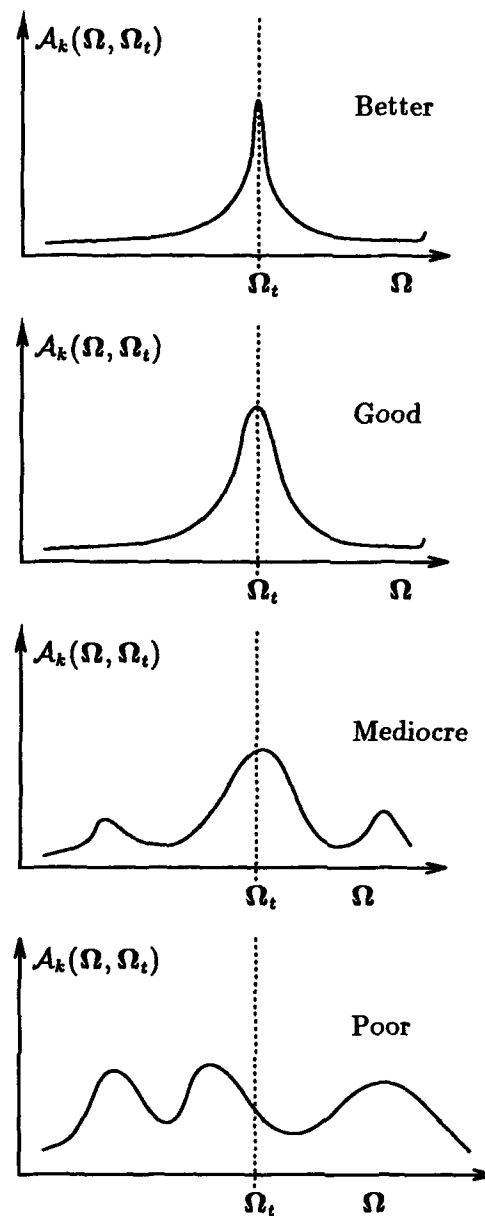


Figure 2.10. Ambiguity Functions for Estimators of Varied Quality

The "mediocre" ambiguity function may be so rated because of any of several non-ideal factors. First, it has peaks other than the one at  $\Omega_t$ , so that a likelihood function parameterized for the wrong values may give a response close to the likelihood function parameterized for the correct value. This may lead to difficulty in making a correct decision as to which true but unknown state/parameter set is present. Second, its main peak does not have a maximum value exactly at  $\Omega_t$ . Third, this ambiguity function has slower rolloff than the "good" function. Finally, the "poor" function reflects the undesirable qualities of the mediocre function to a particularly exaggerated degree.

Furthermore, particularly for applications where we will be able to extract only a limited number of likelihood function values prior to making a state/parameter identification, we desire that the likelihood function values for any combination of  $\Omega$  and  $\Omega_t$  be closely distributed about the mean or ambiguity function value. This need drives us to consider determining not only the ambiguity function, but also higher level moments of the likelihood function *probability density function* for likely combinations of  $\Omega$  and  $\Omega_t$ . Ambiguity functions can be developed analytically for some likelihood functions [152, 154], or in any case empirically by Monte Carlo simulation runs or extended experimentation. Likewise, likelihood function probability densities might presumably be found analytically in some cases (as an extension to [152], for example), but could also be estimated by Monte Carlo or experimental research.

To relate these curves to the classical "probabilities of correct/incorrect recognition", etc., generally used in ATR, note that a horizontal "threshold" line could be drawn through these curves. The likelihood function probability density functions (recall that the generalized ambiguity function is the mean) for the true parameter point (object) and some other parameter point (object) then allow one to establish the relative probability of likelihood function values above or below the threshold at either object. This allows an immediate estimate of probability of correct/incorrect recognition, etc., for that threshold. This can only be an estimate of such probabilities, however, because the generalized ambiguity function is a *mean* value over many (ideally, all possible) measurements, and as such, it obscures correlations between responses for individual measurements that could tend to

cause likelihood functions for both of two different objects to fall above or below some threshold.

If a likelihood function is expressed as a natural logarithm (so that the first and second derivatives give a “non-dimensional” or scaled slope and “curvature” of the likelihood function), the curvature of the corresponding ambiguity function near the value of  $\Omega_t$  can be related to the Cramér-Rao lower bound (CRLB) [184] of the covariance for a state/parameter estimate obtained by the use of the underlying likelihood function [154, 198]. For Gaussian likelihood functions, which are naturally expressed in log form, this relationship is particularly powerful.

Cramér-Rao lower bounds and the class of analogous bounds called Cramér-Rao-like lower bounds are of much interest in estimation theory, since they define a theoretical bound on the quality of information that can be inferred from an estimator [125, 34, 210]. For a vector parameter  $\Omega$  estimated by some process yielding an unbiased estimate  $\hat{\Omega}$ , where the true parameter value set is given by  $\Omega_t$ , the Cramér-Rao lower bound on maximum likelihood estimate error variance is given by:

$$E\{[\hat{\Omega} - \Omega_t][\hat{\Omega} - \Omega_t]^T\} \geq \left[ -\frac{\partial^2}{\partial \Omega^2} \mathcal{A}_k(\Omega, \Omega_t) \Big|_{\Omega=\Omega_t} \right]^{-1} \quad (2.38)$$

where  $\mathcal{A}_k(\Omega, \Omega_t)$  is the ambiguity function corresponding to the log likelihood function, and all terms have been defined previously. This expression provides a lower bound as well for biased estimates, for which a “tighter” lower bound can be defined if the bias is well-characterized [154:97].

Recently, interest has been directed toward defining and using Cramér-Rao-like lower bounds for object tracking [60, 62, 61, 39]. No analogous bound has been published for dynamic object or target recognition, and at least one recent article has solicited such bounds on performance [30]. It will be noted in Chapter V that evaluation of object recognition algorithms by generalized ambiguity functions offers a natural extension to the concept of a Cramér-Rao-like lower bound for object recognition, although in fact the ambiguity function or classical probabilities of correct/incorrect recognition may be more

meaningful approaches by which to define recognition performance. Also in Chapter VI, we will discuss the utility of a Cramér-Rao lower bound in object recognition.

In 1979, Altes [1] evidently recognized the applicability of maximum likelihood estimation and a *generalized ambiguity function* (defined rather differently than the concept by Schweppe referred to above) to object recognition, but Altes' proposals have not been exercised in the open press (or in the classified press, so far as can be determined). Subsequent researchers developing Altes' approach [2, 7, 173, 200] have been concerned only with maximum likelihood *state* estimation (as for locating submerged objects with sonar) rather than parameter estimation as for object recognition (note that some of the referenced articles use the term "parameters" to refer to position information which we would term "states").

Aside from the writing of Altes, so far as the author has been able to determine in an extensive survey (including all references listed in the bibliography), neither the concept of *generalized likelihood functions* nor the *generalized ambiguity function* has been employed in multisensor fusion for object recognition, at least in the explicit *estimation theoretic* sense in which they have been employed for estimation of states and parameters [152, 154, 198]. It seems clear, however, that many published systems *could* be analyzed from this perspective – it is, however, the author's firm impression that *they have not been*. This effort has done so.

## 2.8 Conclusion

With the conclusion of this chapter, we have in place all of the tools with which to define and conduct the proposed research. In particular, we have discussed two classes of algorithms – the Kalman filter and related estimators, and dynamic programming sequence comparison – and their previous application in state and parameter identification and syntactic pattern recognition.

In following chapters, we will develop the theory and practice for applying these tools – separately and together – in dynamic object recognition. Our intent in each case will be to exploit for recognition the *characteristic coupling* between states, parameters,

and measurements of dynamic physical objects. We will use knowledge of such coupling and sequences of measurements, evaluating the consistency of measurement sequence generation with the known coupling for each object class. Knowledge of characteristic coupling allows us to consider:

- (1) The *joint likelihood of observed events* over time from known target classes, conditioned on past measurements and *a priori* information for each class.
- (2) The *syntax of observed events* from unclassified objects, by comparison with the syntax of event sequences expected from known target classes. For known physical objects, the process of generating sequences or sequence spaces of expected elements for each class inherently considers the joint or coupled nature of the processes which produce the sequences.
- (3) *Restrictions on the domains of likelihood functions* used to identify known object classes, according to joint or coupled behavior expected over time. We will reject object-class associations that do not fit reasonable restrictions.

These three considerations are simply different ways of making the same statement. The only differences in our application of these considerations from case to case will be driven by the limitations of the available tools. We will see that the Kalman filter and dynamic programming sequence comparison techniques possess a combination of characteristics which allow them together to exploit joint likelihood for objects with linear *and* nonlinear state and measurement spaces – a frequent combination for dynamic physical objects.

The remainder of this dissertation will refer to the concepts discussed in this chapter to put the author's research into perspective with previous efforts. The next chapter lays out the major functional elements of the author's contribution. Subsequent chapters will demonstrate those elements.

### III. Exploiting Joint Likelihood in Object Recognition

#### 3.1 Introduction

In this chapter, we consider approaches for exploiting the *joint likelihood* of “kinematic” (classically, position and velocity) and “nonkinematic” (sensor signature or feature observable) measurements for dynamic object and target recognition. We will examine conventional and unconventional estimator structures, and define constructs whereby these structures can be combined to perform specific tasks in particular situations. All of these estimators will involve forms of *sequence comparison*, since measurements from an unclassified physical object occur naturally over time in sequences that contain much information about the joint likelihood of their generation by any particular *a priori* known object class.

Recall that our objective is to make improved (more probably correct) estimates of object class, based or conditioned on measurements from all available sensors. Following the discussion in Sect. 2.6.2, we aspire to produce a pattern recognition system that gives the *a posteriori* probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  that we are actually observing an object of class  $\omega_i$ , given a set of  $k$  *feature observable* measurements  $\mathbf{Z}_k^f = \{z_1^f, z_2^f, z_3^f, \dots, z_k^f\}$ ,  $m$  kinematic or *dynamic* measurements  $\mathbf{Z}_m^d = \{z_1^d, z_2^d, z_3^d, \dots, z_m^d\}$ , and *a priori* object class probabilities  $p(\omega_i)$  for each of  $J$  known object classes. Historically in general, however, as noted in Sects. 2.2 and 2.6.2, only feature observable (“nonkinematic”) information  $\mathbf{Z}_k^f$  has been used for object recognition, although kinematic information has been used to limit search windows.

Expanding significantly the approach of Therrien [211], we will not be content to limit ourselves to cases in which it is possible to make a linear prediction of future measurements, based on current state estimates derived from previous measurements. For feature spaces which are highly *nonlinear* functions of an underlying *dynamic* state space subject to “high frequency” variations or unpredictable transitions (radar signature as a function of aspect angle, in particular), linear or linearized prediction is impossible in a practical sense. Therefore, we now set out to extend the status quo of dynamic object / tactical target recognition in three steps.



First, we will define a new class of recognition algorithms based on kinematic/aspect trackers (recalling Sect. 2.3.3) that combine kinematic information while obeying feature observable constraints. These algorithms will exploit the joint likelihood of observed kinematics, conditioned on feature observable or signature measurements. The results will provide major practical extensions to the proposals of Therrien and Eagle (the latter being discussed previously in Sect. 2.3.3.1).

Second, we will define a different class of algorithms that combine feature observable information while obeying feasible or observed object kinematic constraints. These algorithms will exploit the joint likelihood of measured feature observables, conditioned on kinematic measurements. The results will provide significant theoretical and practical extensions to the efforts of Le Chevalier *et al.* [136] and Mieras *et al.* [164, 165].

Either of these algorithm classes can be used for stand-alone object recognition, and that is a primary approach taken in this research. Third, however, we will show how these two approaches can be combined to yield a *new* estimator structure that uses the joint likelihood of kinematic *and* feature observable observations, conditioned on previous measurements from both domains and other *a priori* information, for real-time recognition of dynamic objects. This estimator structure shows considerable promise for efficient state and parameter estimation in cases involving both linear and nonlinear state space/measurement space relationships. Although timelines for this research did not permit implementation and testing of the third construct, the theory and structure for design of such a new estimator are laid out clearly. All of these developments will comply with well-understood practices of Bayesian parameter estimation.

The next two sections discuss terminology and factors that bear on measurements and estimation for dynamic objects in general and tactical targets in particular. Subsequent sections address new recognition schemes. The last section in this chapter motivates new techniques for evaluating algorithm performance. Techniques considered include classical probabilities of correct and incorrect identification, as well as the generalized ambiguity function introduced in the previous chapter.

### 3.2 Spaces and Dimensions.

In the subsequent discussions, it will be necessary to deal with at least four different concepts of spaces and their associated dimensions. All have been previously introduced in this dissertation, but their subsequent close association may cause some confusion, so they are revisited here for contrast and comparison. These are (1) physical or three-space, (2) aspect angle space, (3) feature or feature observable space, and (4) warping path space.

**3.2.1 Physical or Three-Space.** This is the usual Euclidean representation of three dimensional physical space. In general, the behavior of real objects in this space is described by positions and higher derivatives in six degrees-of-freedom ("6-DOF"), corresponding to three translational and three rotational degrees of freedom – translation and rotation states. Behavior can only be quantified relative to some reference frame, generally Cartesian, which may be stationary or non-stationary in physical space. Position or displacement in the rotational degrees of freedom is referred to as angular orientation.

It is important to note that for a 6-DOF object, the term *kinematics* properly refers to both translational *and* rotational state behavior. As we have noted in previous chapters, however, the general inability to measure rotation state directly using remote sensors has led to the use of the term " 'kinematic' measurements" to refer to measurements of translational state variables only – classically, position and velocity. In this development, the distinction will be clarified where required.

**3.2.2 Aspect Angle Space.** This space is the entire  $4\pi$  steradian extent of the hypothetical aspect angle sphere (equivalently, the surface of the unit sphere) as shown in Fig. 1.2, which is closed under allowable transitions on that sphere, and which is therefore considered an aspect angle "space." It may also be necessary to speak of a region or "window" on, or subset of, the hypothetical aspect angle sphere. The entire aspect angle space and regions in general are inherently two-dimensional, if we consider only in-plane rotation-invariant (see "PSRI," in App. A) feature observables. These cases may require specification of feature observable values as a function of object-frame relative azimuth and

elevation angles, as shown in Fig. 5.1, or as a function of the equivalent object-to-sensor unit vector in the object body frame.

In dealing with a non-rotation-invariant feature observable, or where a transformation is required from aspect angle to object rotation state with respect to some frame in physical space (discussed below), a third dimension – roll about the object-sensor vector – must be added to specifications of aspect angle. An aspect angle path, as seen in Fig. 1.2, in azimuth and elevation angle (and possibly roll, although not visible in the figure) is a one-dimensional aspect angle region or subset.

Henceforth, a particular aspect angle state or location in aspect angle state space (continuous or discretized) will be denoted  $\mathbf{x}^a$ . The superscript  $a$  refers to aspect angle, just as superscript  $d$  denotes translational states (associated with “kinematic” or *dynamic* measurements). In general, each value  $\mathbf{x}^a$  represents the orientation of the observing sensor relative to the observed object or target body frame at some particular time of interest. For some time  $t_k$ , the corresponding aspect angle state is given as  $\mathbf{x}_k^a$ .

For a known sensor angular orientation relative to some reference frame in physical space, three-dimensional aspect angle is equivalent to a sensor-relative representation of the object’s angular orientation or “rotation state” relative to that physical space reference frame. Since this restriction will apply in general for the discussions here, the term  $\mathbf{x}^a$  may be used interchangeably for sensor-relative “aspect angle” or rotation state relative to some other, generally stationary reference frame in three-space. Distinctions between aspect angle state relative to the sensor and rotation state relative to some other frame will be clarified where necessary.

**3.2.3 Feature or Feature Observable Space.** This is the classic feature space discussed in Sect. 2.2 – a vector space in which a given feature observation or measurement of the object can be expressed as a point. Thus, the dimension of the feature space is determined by the number of individual scalar elements in a feature observable measurement vector. The dimensions may or may not be independent in any particular sense of the word – functional, statistical, and/or temporal relationships may exist between feature space “dimensions”. Feature spaces of dimension  $n$  are generally taken to be isomor-

phic [170:173] to  $R^n$ , the (Cartesian) vector space defined by a collection of  $n$  mutually orthogonal real axes sharing a common origin.

The Cartesian product space formed by the aspect angle space and some feature space provides a framework for description of the appearance of any object to a sensor system extracting the given features as the object traverses any given path in aspect angle. Each object model corresponds to a particular set of feature observable values for each aspect angle value in this Cartesian product space. Due to the stochastic nature of feature observable generation and measurement, a given point in aspect angle space for any one model will invariably be associated with many points in feature observable space. The realization likelihood of particular feature observable values for any aspect angle will be governed by some probability density, which generally will be dependent on the state of the object, sensor, and other factors (consider, for example, the effect of tactical target engine vibrations on radar returns).

As in Sect. 2.6.2, a particular element (generally, a noisy measurement or observation) from a feature observable space (continuous or discretized) will be denoted  $z^f$ , where the superscript  $f$  refers to feature. A sequence or vector composed of such elements, where in general each element corresponds to a different measurement time, will be denoted  $Z^f$ , as in Sect. 2.6.2.

**3.2.4 Space of Allowable Warping Paths.** As defined and discussed in Sect. 2.4.3, this is the set of all possible associations between elements in two sequences, or between a sequence and a region from which sequences can be extracted. When comparing a one-dimensional sequence to another one-dimensional sequence, the space of all possible associations is two-dimensional, as shown in Fig. 2.8. When comparing a one-dimensional sequence to a two-dimensional set of possible sequence elements, the space of all possible associations is three-dimensional, as shown in Fig. 2.9.

### 3.3 Joint Likelihood in Object Recognition

Recall the discussion of multiple model filters and parameter identification in Chapter II (see Eqns. (2.20) through (2.22)), following the development in Maybeck Vol. II

[154:129–136]. We noted that conventional (e.g., linear, linearized or extended Kalman) filter structures can be defined readily in a Bayesian, multiple model *residual sequence analysis* framework to perform parameter estimation.

In that conventional multiple model parameter identification approach, we use the *a priori* likelihood of each model (feasible parameter set) and – from residual analysis – the *joint likelihood* of each model having produced the observed measurements at each measurement event over time to identify the most probably correct parameter set. We noted that this approach is intimately related to the target recognition approach taken by Therrien [211], and recommended by Daum [9:177–178]. In target recognition, the discrete sets of parameters are the known target classes, and we have measurements in “kinematic” (physical translation space) and feature observable domains.

For many object or target states, parameters, and measurements of interest, a conventional filter will adequately model the desired object behavior, and provide a basis for residual analysis; that is, checking the deviations over time between (1) measurements *predicted* for each object class, and (2) measurements *observed* from the unclassified object. This is particularly true for the classical “kinematic” or translational states and associated measurements of range, pointing angles, and associated rates, or alternative physical space equivalents.

But physical objects are inherently six degree-of-freedom or “6-DOF” entities – that is, to describe their kinematics or motion fully, we need to consider states (positions and derivatives) in three translational *and* three rotational state subspaces. A proper description would in general consider each subspace separately, but the state dynamics for many physical objects are highly *coupled* across the subspace boundaries. For any particular object, the nature and degree of this coupling are generally a function of object parameters that can be directly or indirectly expressed in the state dynamics (mathematical) model.

For example, in the case of conventional airplanes (recall Sect. 2.3.3.1), rotational states are often closely coupled to translational acceleration states. The parameters which govern this coupling are aircraft class-specific quantities like wing surface area, coefficient of lift, mass, and so on. The coupling implies that rotation state measurements could be

tray much regarding future kinematic velocity and position. Unfortunately, the kinematic position measurements to which most object trackers are limited provide little observability of rotation state (or aspect angle, from a sensor-relative perspective).

Now for most objects of interest, translational and rotational dynamic states are related in a nonlinear fashion, but often can be linearized for modelling and prediction in a conventional filter structure. If we could measure translational and rotational states *directly* for some class of objects and apply these measurements to an appropriate multiple model set of conventional filters, we might expect that an *incorrect* combination of measurements and filter model would exhibit high residual error, where the true and filter model state *dynamics coupling assumptions* differ. This high residual error would be an expression that *that* particular filter model has a low *joint likelihood* of producing the observed translational *and* rotational measurements. The residuals could then be combined in the classical way to yield an *a posteriori* probability of class membership.

In general, it is very simple to obtain "kinematic" or translational measurements that are linear functions of, or readily linearized with respect to, translational state spaces. Unfortunately, as noted above, we cannot in general measure "rotation state" directly for remotely observed objects. However, feature observables or signatures are generally direct functions of aspect angle, and their measurements potentially contain much information about rotational states.

The problem is that, in typical cases of interest, conventional estimators often cannot exploit the relationship between *rotational states* and *feature observable measurements*. Simply, it is frequently impossible to make a reasonable linear or linearized prediction of the expected feature observable measurement from current knowledge of aspect angle or rotational state. Under these circumstances, the classical approach of Therrien [211] or multiple model residual analysis cannot be used.

First, particularly for aircraft targets, true aspect angle state values are likely to change unpredictably over time. In a classical state estimation sense, we say that they have high "driving noise strength" or "*Q*" (see Sect. 2.3.1), driven directly as they are by unobservable, unpredictable, often high gain operator inputs (e.g., roll/pitch commands).

Second, the relationship between feature observable measurements  $\mathbf{z}^f = \mathbf{h}(\mathbf{x}^a) + \mathbf{v}$  (relaxing the notation from Sect. 2.3.1.1 somewhat for clarity) and aspect angle state  $\mathbf{x}^a$  may be poorly understood or highly nonlinear, even to the extent that the partial derivative  $\partial \mathbf{h}(\mathbf{x}^a)/\partial \mathbf{x}^a$  and expected measurement  $\hat{\mathbf{z}}^f = \mathbf{h}(\hat{\mathbf{x}}^a)$  information required by linear or linearized filters (Eqn. (2.18)) cannot be reasonably defined.

Even if the relationship is known, the “deterministic” measurement component  $\mathbf{h}(\mathbf{x}^a)$  of  $\mathbf{z}^f$  (see Eqn. (2.15)) and its partial derivative may vary so quickly with respect to aspect angle state that reasonable, otherwise unobservable changes in aspect angle render the partial derivative and expected measurement unpredictable. In other words, if the unpredictable aspect angle state change between measurements (a function of  $\mathbf{Q}$  and inter-measurement time interval) is likely to be such that the expected measurement  $\mathbf{h}(\hat{\mathbf{x}}^a)$  cannot be predicted, or the variation of  $\mathbf{h}(\mathbf{x}^a)$  cannot adequately be approximated as linear over the range of current and likely future  $\mathbf{x}^a$  values, then conventional “predictor-corrector” filtering won’t work.

This is precisely the situation for aircraft targets and their radar signatures (particularly narrowband, non-high range resolution) due to high aspect angle rates and scatterer interactions as described in Chapter II. We can illustrate this problem for any notional feature space (*not limited to radar*), as shown in Fig. 3.1. Suppose your state location at time  $t_0$  is somewhere in the region marked  $\mathbf{x}_0^a$ , but at the next measurement time  $t_1$ , when your state may lie somewhere in the region marked  $\mathbf{x}_1^a$ , you receive a measurement  $\mathbf{h}(\mathbf{x}_1^a)$  (a function solely of some unknown particular  $\mathbf{x}_1^a$ ). As a conventional filter designer, your first questions are (1) “what value of  $\partial \mathbf{h}(\mathbf{x}^a)/\partial \mathbf{x}^a$  is appropriate?” and (2) “what is my expected measurement  $\hat{\mathbf{z}}_1^f = \mathbf{h}(\hat{\mathbf{x}}_1^a)$  at  $t_1$ ?”. We quickly observe that, due to the rapid variations of  $\mathbf{h}(\mathbf{x}^a)$  as a function of  $\mathbf{x}^a$  and the uncertainty in your current and new state locations, it is clearly *impossible* to define a meaningful partial derivative value or an expected measurement.

The conditions put forth in the scenario of Fig. 3.1 are particularly stressing. The next section will examine conditions under which conventional multiple model residual analysis techniques *can* be used with kinematic and feature observable information for

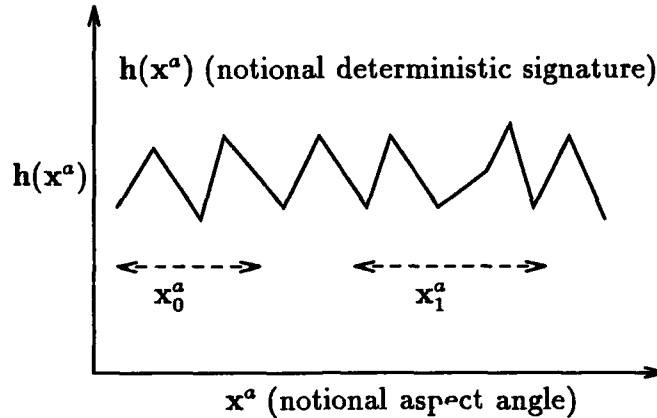


Figure 3.1. A Measurement Function Unsuitable for Conventional Filtering

dynamic object / target recognition – *without* obtaining a linear predictive model for feature observable measurements.

### 3.4 Conventional Multiple Model Approaches for Dynamic Object Recognition

In this discussion, we will limit ourselves to filter state dynamics models that include *coupled* translational and rotational (or aspect angle) states only. We will exploit this coupling for multiple model residual analysis while using – but not predicting – feature observable measurements. The outcome of this process will be an expression for the joint likelihood of translation state-related or “kinematic” measurements and rotation state-related *pseudo-measurements*, conditioned on feature observable measurements. This is Step One in the three step process outlined at the start of this chapter.

**3.4.1 Required Conditions and Estimator Alternatives.** First, suppose that a one-to-one (see App. A) mapping exists between aspect angle space and some feature observable space, and is usable as a transformation between the two spaces, for a particular object of class “A” (failing the one-to-one requirement, we require that the inverse mapping is of “low ambiguity”, in the sense of Sect. 2.2.1). This mapping is said to provide a *pseudo-measurement* of aspect angle, or equivalently a *pose estimate*, as defined in Sect. 2.2.1. For the *correct* association of “kinematic” (translation state) measurements



and aspect angle pseudo-measurements from this object with its coupled dynamics filter model, a feature observable measurement will be equivalent to an aspect angle or rotation state measurement (noting the usual relationship between aspect angle and rotation states discussed in Sect. 3.2.2). Moreover, we expect that the residual sequence (i.e., from all 6-DOF kinematic measurements and pseudo-measurements) for this correct association should indicate high *joint likelihood* of the observed "measurements".

However, suppose that the aspect angle-to-feature observable mapping and/or dynamics coupling is different for some other object of class "B". In this case, we should *not* expect that the aspect angle pseudo-measurement derived by mapping feature observations from class "A" into the aspect angle space of class "B" will lead to high likelihood (small) residuals from the filter model corresponding to class "B".

Thus, we have a set of conditions under which the joint likelihood of all 6-DOF kinematic events (translational *and* rotational measurements or pseudo-measurements) can be assessed, conditioned on prior kinematic *and* feature observable measurements, without *explicitly* considering the likelihood (or probabilistically-weighted distance in feature space) of the feature observable measurements *per se*. Restated, those conditions are (1) existence and availability of a low-ambiguity aspect angle-to-feature observable mapping for each class, (2) differences in this mapping for different object classes, and (3) differences in the state dynamics coupling for different classes. Condition (1) must apply in every case, but either or both of (2) and (3) must apply only to the extent required to obtain distinctly different residual sequences or other evidence of incorrect associations for objects and behavior of interest.

Observe in passing that the aspect angle-to-feature observable mapping need neither be linear nor "onto" (see App. A) the entire feature observable space (i.e., the mapping must only be onto the range of the transformation for each class). The point here is that a nonlinear mapping can still give us the one-to-one condition that we require, and a mapping that is not "onto" the feature space allows us to reject outright the need to investigate a given class which cannot map from its aspect angle space to the measured feature observable value.

For tactical targets, the Kendrick-Maybeck / Andrisani / Sworder-type "kinematic/aspect angle" estimators provide exactly the sort of coupled dynamics filter model that we seek. They use "kinematic" (translation state) measurements in conventional linear ways, and ignore the detailed relationship between true aspect angle and feature observable measurements by allowing the estimator to accept only a pseudo-measurement of aspect angle, generally from some pose estimator, rather than the feature observable measurement itself. In the following section, we consider exactly how a kinematic/aspect-angle filter would be used in a multiple model estimator for object recognition.

**3.4.2 Kinematic/Aspect-Angle Estimators for Recognition.** We assume that conventional measurements of object position in three space and relative (doppler) velocity along the object-sensor vector are available at discrete times (measurement intervals). Moreover, we assume that pseudo-measurements of object aspect angle are available at the same times, say in the form of three independent Euler angles relative to some vector frame of reference. In the nomenclature of Section 2.3.1.3, this defines a seven-state  $\mathbf{z}_k$  vector, which provides an update to each of  $J$  kinematic/aspect-angle filters (one for each candidate object class) at each measurement event.

Recall that the pseudo-measurements of object aspect angle  $\mathbf{z}^a$  are in fact based on aspect angle or pose estimates  $\hat{\mathbf{x}}^a$  (the "hat" denoting an estimate). In general, we have  $\mathbf{z}^a = \mathbf{h}(\hat{\mathbf{x}}^a)$ , with  $\hat{\mathbf{x}}^a = \arg_{\mathbf{x}^a} \{\max[p(\mathbf{z}^f | \mathbf{x}^a, \omega_i)]\}$ . Depending on the representation for object rotational states in the kinematic/aspect-angle filter, the function  $\mathbf{h}$  and filter measurement matrix  $\mathbf{H} = \partial \mathbf{h}(\mathbf{x}^a) / \partial \mathbf{x}^a$  are at most simple coordinate transforms, and can be identity operations. Thus, in fact, the aspect angle pseudo-measurement contains information provided by feature observables over time, or  $\mathbf{Z}^f$ . The measurement covariance for the pose estimate is a function of the *a priori*-characterized performance for the pose estimator of each object class, for the given ambient conditions.

The object position and velocity measurements (common to all object classes) and aspect angle pseudo-measurements (in general, unique to each class) are provided to the appropriate set of kinematic/aspect-angle filters – one filter for each object class. In turn, the filters generate translation state- and rotation state-related (i.e., coupled 6-DOF

kinematics) residuals and covariances. We can now follow Sect. 2.3.1.3 to define an object classifier using residual analysis for "kinematic" (translation) measurement and "aspect angle" pseudo-measurement residuals in the following Bayesian structure:

$$p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_{k+s}^d) = \frac{\left\{ \prod_{n=1}^k [p(\mathbf{z}_n^{d,a} | \mathbf{Z}_{n+s-1}^d, \mathbf{Z}_{n-1}^f, \omega_i)] \right\} p(\omega_i)}{\sum_{\omega_j, j=1}^J \left\{ \prod_{n=1}^k [p(\mathbf{z}_n^{d,a} | \mathbf{Z}_{n+s-1}^d, \mathbf{Z}_{n-1}^f, \omega_j)] \right\} p(\omega_j)} \quad (3.1)$$

where:

$\mathbf{Z}_k^f$  = a set of  $k$  feature observable measurements  $\mathbf{z}^f$

$\mathbf{Z}_{k+s}^d$  = a set of  $k + s$  "kinematic" (translation state) measurements  $\mathbf{z}^d$ , where  $s \geq 1$  in general (representing kinematic measurements which precede the  $k$  feature observable measurements – to be discussed in Sect. 3.6.5)

$\mathbf{z}_n^{d,a}$  = a vector formed by juxtaposing "kinematic" measurements  $\mathbf{z}^d$  and aspect angle (pseudo) measurements  $\mathbf{z}^a$  taken or available at time  $t_n$

and other variables are as defined earlier.

An immediate extension to this concept is to allow "state reasonableness monitoring" as discussed in Sect. 2.3.1.3. Under this construct, our Bayesian expression is of the form:

$$p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_{k+s}^d) = \frac{\left\{ \prod_{n=1}^k [p(\mathbf{z}_n^{d,a}, \hat{\mathbf{x}}_n^r | \mathbf{Z}_{n+s-1}^d, \mathbf{Z}_{n-1}^f, \omega_i)] \right\} p(\omega_i)}{\sum_{\omega_j, j=1}^J \left\{ \prod_{n=1}^k [p(\mathbf{z}_n^{d,a}, \hat{\mathbf{x}}_n^r | \mathbf{Z}_{n+s-1}^d, \mathbf{Z}_{n-1}^f, \omega_j)] \right\} p(\omega_j)} \quad (3.2)$$

where:

$\hat{\mathbf{x}}_n^r$  = an estimated state or states which we wish to compare to anticipated values for each model  $\omega_i$  at each time  $t_n$

and other variables are as defined earlier.

**3.4.3 Summary.** Given a pose estimate from a feature observable sensor and associated pose estimators, then, Kendrick *et al.*-type filters in a multiple model structure provide a natural tool for residual analysis and Bayesian estimation of object class mem-

bership, considering the *joint likelihood* of object behavior with respect to kinematic states and measurements that the filter is well-suited to process. Chapter IV will propose a suitable estimator model for implementing these equations, and show how residual modeling and state reasonableness checking betray an incorrect object-model association.

This straightforward, conventional multiple model construct provides Step One in the three-step process discussed at the start of the chapter. The second (most challenging) and third steps remain to be solved. In the following section, we pursue an alternative approach to object recognition, which will lead to Step Two. Those results will be integrated ultimately with conventional multiple model techniques to form a new class of estimators – satisfying Step Three.

### 3.5 Comparing Observed and Expected Feature Observables Without Linear Prediction

Recall from Sect. 3.4.1 that necessary conditions (2) and (3) for the use of kinematic/aspect filters in object recognition required differences in the feature observable-to-aspect angle mapping for different objects, and/or differences in the state dynamics coupling for different classes. The reader should note that these conditions are not *sufficient* for the development of residual differences between correct (true) and incorrect filter models. That is, we may conceive of some true object and trajectory which yield, from some feature observable domain, pose estimates that are reasonable for both correct and (one or more) incorrect object library classes, given the observed kinematics.

Kinematically then, even with a pose estimate, the two classes are ambiguous, and to distinguish them we may need to look also at matching distances in the *feature observable space*. Recall that the pose estimate is an angle value, and it says nothing about the closeness of measured and library signatures in feature observable space – a single signature could yield reasonable pose estimates for two different library classes, even though the matching distance *in feature observable space* is much greater for one library class than for the other.

Anticipating the need to recognize dynamic objects with ambiguous feature observable spaces, however, as implied in the introduction to Sect. 3.3, we are still not confident

to make a predictive estimate (linear or otherwise) of the next feature observable measurement, for comparison to the observed measurement. What other alternatives do we have for comparing the observed feature observable measurement to possible values for each object class?

First, for a discretized aspect angle space  $\mathbf{x}^a$ , a typical decision theoretic, parametric pose estimator for object class  $\omega_i$  may provide the value  $\max_{\mathbf{x}^a} \{p(\mathbf{z}^f | \mathbf{x}^a, \omega_i)\}$  – that is, the likelihood of observing a *single* feature observable value  $\mathbf{z}^f$  at the aspect angle value  $\mathbf{x}^a$  (the pose estimate) where that likelihood is maximized. Note the use here of a lower case  $\mathbf{z}$ , referring to a measurement at a *single* discrete time. For a statistically well-behaved pose estimator, that is, one exhibiting low, unbiased ambiguity with respect to aspect angle for the correct object-library association, this output likelihood may be a useful metric for comparison.

In general, however, comparing these values for two different object classes  $\omega_i$  and  $\omega_j$  is a questionable approach to comparing distances in feature observable space, because each such value represents only a maximum likelihood value derived from *one* aspect angle state, rather than a maximum *a posteriori* estimate found by considering all possible states  $\mathbf{x}^a$ . Under circumstances where the likelihood of a given feature observable measurement changes rapidly with aspect angle – as seen in Fig. 3.1 – use of the likelihood from the pose estimator is decidedly suboptimal. Moreover, in these cases we expect that a classical pose estimate history – considering each feature observable measurement independently – will be very “noisy”, exhibiting random changes in aspect angle, perhaps even for the correct object-library association. How can we derive an accurate aspect angle estimate for the (unknown *a priori*) correct class in these cases?

To address these issues, we will develop the methodology of Larson and Peschon for use in object recognition. It will be seen that the Larson and Peschon approach is eminently applicable for dealing with aspect angle states and ambiguous feature observables, provided that we have some basic knowledge about the “*a priori*” likelihood of transitions in the aspect angle space  $\mathbf{x}^a$ .

Unlike conventional estimators, the Larson and Peschon estimator does not create *one* predicted state vector and corresponding predicted measurement for comparison with the next measurement. Rather, the Larson and Peschon estimator evaluates *all* reasonable new state vectors (candidate locations), considering the likelihood of observing the new measurement at each candidate location, and the likelihood of being at that new location given (1) the likelihood of being at some previous state location, and (2) the likelihood of moving from that previous location to the current candidate location. The use of dynamic programming provides a method for performing these calculations without resorting to an exhaustive search covering all state locations over all time.

Finally, as we have seen, the output of the Larson and Peschon equations is naturally a measure of the maximum *joint likelihood* of observed events, conditioned on *a priori* information. For this reason, it will fit naturally into our desire to exploit the joint likelihood of observed events, allowing us to work with aspect angle and feature observable spaces that cannot be treated with conventional linear estimation techniques.

Sect. 3.6 develops the Larson and Peschon methodology in detail for object recognition. We have seen in Sect. 2.4.2 that the Larson and Peschon equations are a particular form of forward dynamic programming sequence comparison – this observation will lead naturally to the consideration of classical dynamic programming sequence comparison and other sequence comparison methods for dynamic object recognition.

Development of these forward dynamic programming sequence comparison-based techniques for dynamic object or target recognition will accomplish Step Two of the three step process outlined at the start of this chapter. This step will be accomplished by looking at differences between objects in feature observable space only, but considering likely kinematics. That is, we will use “kinematic” (i.e., translation state-derived) measurements, with knowledge of the state dynamics translation/rotation coupling for different object classes, to *restrict* our choices for matching feature observable measurements to feature observable libraries for each class according to likely dynamics. The result will be an expression for the joint likelihood of measured feature observables for each class, conditioned on kinematic measurements.

This process will not require differences in the translation/rotation state coupling between classes. However, as a consequence of this process, a high quality pose estimate will be obtained for each class, even for noisy feature observable spaces. This fact will allow us to combine kinematic/aspect recognizers with forward dynamic programming-based recognizers for use in ambiguous feature observable spaces. This combination will accomplish Step Three of the three step process. The result is a new form of estimator.

The following section develops extensions to the Larson and Peschon equations for object recognition. Subsequent sections consider other forward dynamic programming sequence comparison-based approaches – both new forms developed in this research and related concepts developed by other researchers. Performance of these recognizers is demonstrated in Chapter V.

### 3.6 Applying the Larson and Peschon Approach in Recognition

The purpose of this section is to apply Bayes' Rule [153, 197], the Larson and Peschon methodology (see Sect. 2.4.4 and [133]) and aspect angle state transition information given by kinematic state estimates (i.e., 6-DOF kinematic state estimates, possibly based on translational measurements only: see Sect. 5.5), to provide a representation for the *a posteriori* probability of class membership  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  for each class  $\omega_i$ , or a reasonable estimate thereof. In this development, the new information, in a Bayesian sense, is given by the feature observable measurements. The *a priori* information is given by kinematic measurements and prior knowledge of class probability  $p(\omega_i)$ .

This effort will achieve Step Two of the three-step process outlined at the start of this chapter, providing an estimator structure that handles dynamic, nonlinear feature observable / aspect angle relationships in a straightforward Bayesian fashion. The development in this section will show that, in general, calculation of the *a posteriori* probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  requires an exhaustive search, but that use of the Larson and Peschon approach allows one feasibly to define a "best estimate" for that quantity. Recalling the discussion in Sect. 2.6.2, we will observe in Sect. 3.8 that this Larson and Peschon approach to fusion of kinematic and nonkinematic or feature observable information provides a more

optimal approach for restriction of the domain of a feature observable-matching likelihood function than prior efforts by Le Chevalier and Mieras, as discussed in Sect. 2.5.

The reader should note that the efforts discussed here are believed to be both original and significant. The contributions are:

- (1) The first application of the full Larson and Peschon approach to dynamic object recognition (as opposed to previous suboptimal applications of the Larson and Peschon equations, or functionally similar approaches, previously described in Sect. 2.5, and further compared in Sect. 3.8, to follow).
- (2) The first known theoretical procedure for obtaining maximum *a posteriori* (MAP) probability of class membership based on feature observable *and* kinematic measurements, and *a priori* probability of class membership, where linear (or linearization-based) estimation techniques cannot be used.
- (3) Suboptimal but practical developments of contribution (2) using outputs of the Larson and Peschon equations to approximate the proper *a posteriori* probability to any desired confidence level.

**3.6.1 Facts and Assumptions.** First, we confine our attention to a set of  $J$  *a priori* known object classes  $\omega_i$ , each represented by a hypothetical aspect angle sphere (see Figs. 1.2 and/or 2.6) or model having appropriate feature observable distributions associated with each aspect angle value. Now, for any given problem (i.e., any given set of discrete measurements  $\{\mathbf{Z}_k^f, \mathbf{Z}_m^d\}$  over some time interval), we can restrict our concern to a given aspect angle region on each of the object spheres – that is, we assume a negligible probability that the class has presented aspect angles outside this region over the duration of the time interval corresponding to measurements  $\mathbf{Z}_k^f$ .

The aspect angle region can be thought of as the union of a set of aspect angle “windows”, each of which is a “sub-region” of feasible aspect angle corresponding to a particular measurement  $\mathbf{z}^f$  in the sequence  $\mathbf{Z}_k^f$ . Fig. 3.2 shows how these windows relate to the region as a whole. The vertical separation between windows is for clarity only.



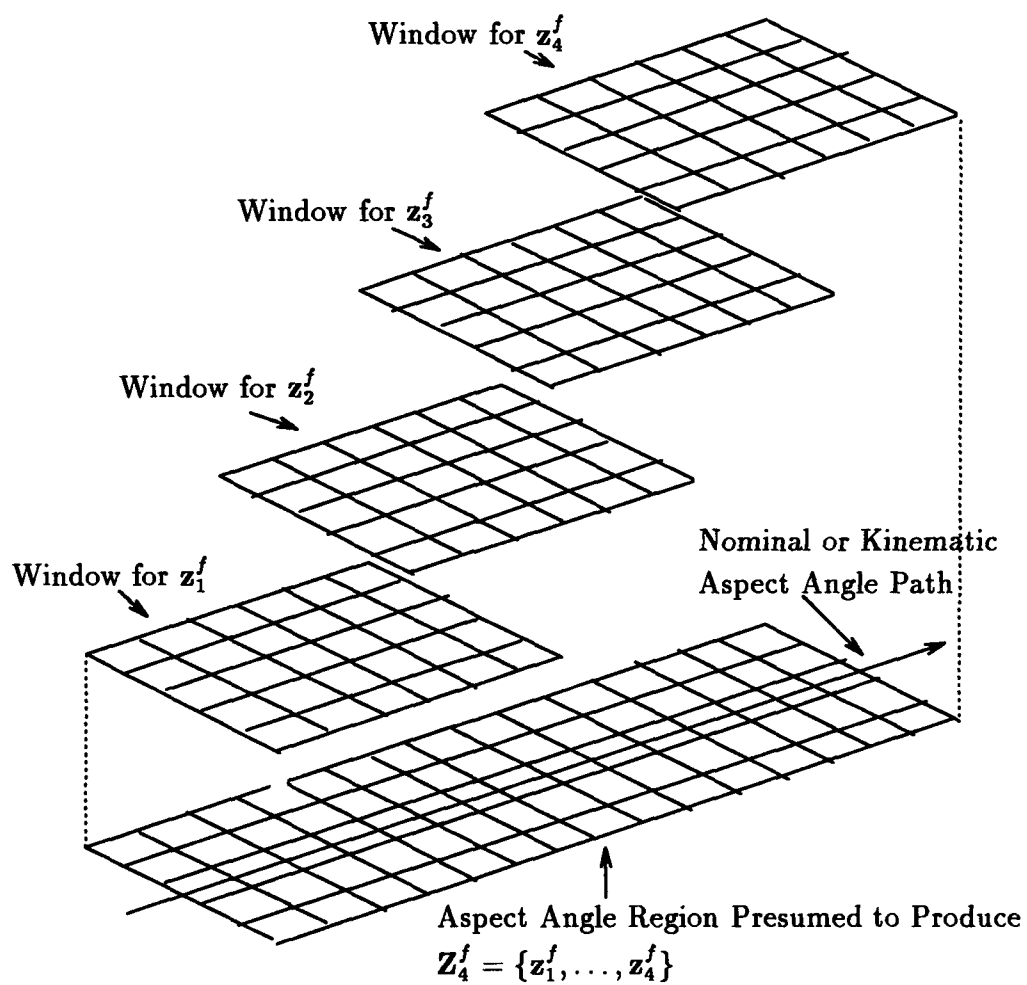


Figure 3.2. Aspect Angle Windows and the Nominal Aspect Angle Path – All Algorithms

Note that the time interval corresponding to kinematic measurements  $\mathbf{Z}_m^d$  will overlap completely and contain the time interval corresponding to the feature observable measurements. For reasons to be clarified in Sect. 3.6.5, in contrast to the  $k + s$  kinematic measurements discussed in Sect. 3.4.2, the  $m$  kinematic measurements discussed here may in general include measurements taken *after* as well as before the  $k$  feature observable measurements. Also, due to differences in angles of attack, etc., for given maneuvers from different classes, the regions or windows may not be identical from class to class. Given some "granularity" or discretization of continuous aspect angle into "cells" on the object spheres (assumed identical for spheres of all object classes), each cell is considered a state  $\mathbf{x}^a$ .

Now, define the super-region  $\mathcal{X}^a$  as the (super-) set of all aspect angle cells or states that belong to the region of consideration for at least one object class, a total of say  $N_s$  cells or states in number. Any set of  $k + 1$  aspect angle cells, or aspect angle state history, corresponding for analysis purposes to discrete cell locations at feature observable measurement times along an aspect angle path which yields the  $k$  feature observable measurements  $\mathbf{Z}_k^f = \{\mathbf{z}_1^f, \mathbf{z}_2^f, \mathbf{z}_3^f, \dots, \mathbf{z}_k^f\}$ , will be denoted  $\mathbf{X}_k^a = \{\mathbf{x}_0^a, \mathbf{x}_1^a, \mathbf{x}_2^a, \mathbf{x}_3^a, \dots, \mathbf{x}_k^a\}$  (where  $\mathbf{x}_0^a$  is an *a priori* or starting state and the other  $k$  states correspond one-for-one to the feature observable measurements  $\mathbf{Z}_k^f$ ). Note that, as do Larson and Peschon, we choose not to make a measurement at time  $t_0$ , corresponding to state  $\mathbf{x}_0^a$  – this distinction provides a clean boundary between *a priori* information and *new* information.

The (finite) number  $N_p$  of possible such aspect angle paths through  $\mathcal{X}^a$  is given by the standard computation for the number of permutations of  $N_s$  things taken  $k + 1$  at a time, with replacement, or  $(N_s)^{k+1}$  paths. We will denote the set of all such paths as  $\mathcal{X}_{pk}^a$ . Henceforth, this development will use notation somewhat different from that of Larson and Peschon by referring to a particular "nth" path of  $k + 1$  aspect angle states as  $\mathbf{X}_{k,n}^a$  (consistent with the notation  $\omega_i$  referring to an  $i$ th object class).

Clearly, from the definition of  $\mathcal{X}^a$ , some of these paths  $\mathbf{X}_{k,n}^a$  are of negligible probability for one or more object classes, because these paths fall outside the subsets of  $\mathcal{X}^a$  appropriate for those classes. Other paths are of negligible probability for all classes because they correspond to kinematically unlikely or "impossible" aspect angle paths. Since

we are dealing with a finite number of aspect angle cells, we will assign probabilities of starting states and state transitions, rather than the probability *densities* (arising from discretization of continuous space) as discussed by Larson and Peschon – recognizing that, in general, in the limit as discretization becomes finer and the number of states approaches infinity, the distinction is irrelevant.

We will see that applying the Larson and Peschon equations (with reasonable modifications for object recognition) for any one model  $\omega_i$  gives the particular state history  $\mathbf{X}_{k,n}^a$ , call it  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$  (i.e., the “Larson and Peschon” estimate of the aspect angle path for model  $\omega_i$ ), which maximizes the conditional probability  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ . With appropriate modifications, we will be able to find the joint conditional probability  $p(\mathbf{X}_{k,n}^a, \omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ , which we will sum over all possible  $\mathbf{X}_{k,n}^a$  to find the quantity that we desire,  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ . Clearly, it is no longer sufficient to ask what aspect angle path  $\mathbf{X}_{k,n}^a$  we are on – also we must ask over which object model  $\omega_i$  this path is traced. The major point of this section is to understand the relationship between (1) the maximum likelihood path found by the Larson and Peschon approach, i.e.,  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$  for a particular  $\omega_i$ , and a joint conditional probability associated with that path, and (2) the information that we want,  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ .

Further assumptions are:

(1) Following Larson and Peschon, assume that feature observable measurement  $\mathbf{z}_j^f$  is independent of aspect angle state  $\mathbf{x}_i^a$  and  $\mathbf{z}_i^f$  for  $t_j \neq t_i$ . As discussed in Sect. 2.4.4, this assumption is open to question for real world objects and sensors, and is readily relaxed, at the risk of added dimensionality (i.e., computation).

For example, suppose we wish to determine the likelihood of a feature measurement  $\mathbf{z}_i^f$  originating not simply from one aspect angle state  $\mathbf{x}_i^a$ , but originating from a set of states  $\mathbf{x}_{i-1}^a$  to  $\mathbf{x}_i^a$ , passed over between discrete measurement times  $t_{i-1}$  and  $t_i$ . This can occur in high range resolution (HRR) radar when returns from individual pulses are summed over a short time period to create a single sweep or measurement, if the aspect angle is changing over that time. As discussed in Sect. 2.2.3, it is common practice to create one HRR sweep for measurement by summing several dozen pulses over a period of much less than one second. Determination of a measurement likelihood is feasible in this case also, but now

the likelihood " $p(\mathbf{z}_i^f \mid [\mathbf{x}_{i-1,n}^a \rightarrow \mathbf{x}_{i,n}^a], \omega_i)$ " is *path dependent*, and many more cases will have to be considered. For the demonstrations pursued in Chapter V, this issue will not be addressed – all feature observable measurements are assumed to have been generated at a single aspect angle location.

Finally with respect to the issue of measurement independence, we note in passing that each  $\mathbf{z}^f$  will ideally include measurements from stochastically independent sensors *and* feature spaces. This condition will provide further reduction in ambiguity between object classes.

(2) It should be clear, and kept in mind during this development that, for any  $l$ ,  $p(\mathbf{Z}_i^f \mid \mathbf{Z}_{i-1}^f) = p(\mathbf{z}_i^f \mid \mathbf{Z}_{i-1}^f)$ , and analogously that  $p(\mathbf{X}_{i,n}^a \mid \mathbf{X}_{i-1,n}^a) = p(\mathbf{x}_{i,n}^a \mid \mathbf{X}_{i-1,n}^a)$ , as in Larson and Peschon's development.

(3) We assume that the aspect angle state (cell) transition likelihood  $p(\mathbf{x}_{j+1,n}^a \mid \mathbf{x}_{j,n}^a, \mathbf{Z}_m^d, \omega_i)$  and probability of starting cell location  $p(\mathbf{x}_{0,n}^a \mid \mathbf{Z}_m^d, \omega_i)$  are given by some *a priori* knowledge about the likelihood of transitions in the aspect angle space. We will see later in this chapter that there are a wide variety of possible sources for this "*a priori*" knowledge in object recognition scenarios – we will consider various combinations of kinematic and kinematic/aspect tracking filters and smoothers, as discussed in Sect. 2.3. For now we will simply note that this information will be provided primarily by the observed object translational kinematics  $\mathbf{Z}_m^d$ , for a given assumption of object class.

It is *extremely important* to note that the transition likelihood  $p(\mathbf{x}_{j+1,n}^a \mid \mathbf{x}_{j,n}^a, \mathbf{Z}_m^d, \omega_i)$ , fed from external sources in this way, provides a natural path for *optimal* sensor fusion. The recognition and use of that fact is a significant contribution of this research, and also distinguishes this concept from previous efforts that can be derived as suboptimal implementations of the equations developed here. This distinction will be addressed in sections to follow (principally Sect. 3.8.1).

(4) We assume that  $p(\omega_i)$  (*a priori*) is known for each object class  $\omega_i$ , and furthermore that  $p(\omega_i) = p(\omega_i \mid \mathbf{Z}_m^d)$ , that is, that the kinematic measurements and derived kinematic state history provide no information as to the nature of the object. This last assumption is clearly neither true nor desirable, particularly for aircraft recognition when character-

istic trajectories for various aircraft types are classified probabilistically. However, in this development, we wish to assess recognition improvement due to DP sequence comparison methods only, so all objects are considered equally likely to have executed the observed kinematic transitions.

In the event that a particular set of kinematic measurements  $\mathbf{Z}_m^d$  can be associated with a particular event (maneuver), and likelihoods of this event  $p(\mathbf{Z}_m^d | \omega_i)$  are known for each object class, another straightforward application of Bayes' Rule will allow one to estimate  $p(\omega_i | \mathbf{Z}_m^d)$  explicitly as shown in Eqn. (3.3) below. This quantity then replaces  $p(\omega_i)$  in the subsequent development.

$$\hat{p}(\omega_i | \mathbf{Z}_m^d) = \frac{p(\mathbf{Z}_m^d | \omega_i)p(\omega_i)}{\sum_{j=1}^J p(\mathbf{Z}_m^d | \omega_j)p(\omega_j)} \quad (3.3)$$

(5) For high range resolution radar feature observables, we assume that uncertainties in range bin alignment and scale factor are handled by finding  $\text{MAX}\{p(\mathbf{z}^f | \mathbf{x}^a, \omega_i)\}$  for any combination of feature observable measurement  $\mathbf{z}^f$  and trial aspect angle state  $\mathbf{x}^a$  on any model  $\omega_i$ , essentially following the "maximum likelihood" method discussed by Weiss and Friedlander [222], and used by Barniv [8, 13] and others [166].

Note that this maximum likelihood approach should be used with caution for the same reason that we hesitated to use the quantity  $\max_{\mathbf{x}^a}\{p(\mathbf{z}^f | \mathbf{x}^a, \omega_i)\}$  in Sect. 3.3 – for "exotic" (i.e., nonsymmetric, multi-modal, etc.) probability densities, maximum likelihood-based estimates may give significantly different answers from those given by, for example, an ensemble weighted-average probability density (i.e., weighted over all equally possible range alignments, in this case). Similar maximum likelihood assumptions are often made for feature observables other than HRR radar.

*3.6.2 Relating Path Probability  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$  to Object Class Probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ .* Using our current variable notation, recall that Larson and Peschon sought the state history or path  $\mathbf{X}_{k,n}^a$  to maximize  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f)$  in some general state space. Analogously in our case, trying to find a "best" path in aspect angle space over

some model  $\omega_i$ , we might seek a state history to maximize  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ . Following Larson and Peschon's approach, we break this expression out as:

$$p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i) = \frac{p(\mathbf{Z}_k^f | \mathbf{X}_{k,n}^a, \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d, \omega_i) p(\mathbf{X}_{k,n}^a | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d, \omega_i)}{p(\mathbf{Z}_k^f | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d, \omega_i)} \quad (3.4)$$

or, equivalently, by the assumptions above:

$$p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i) = \frac{p(\mathbf{z}_k^f | \mathbf{x}_{k,n}^a, \omega_i) p(\mathbf{x}_{k,n}^a | \mathbf{x}_{k-1,n}^a, \mathbf{Z}_m^d, \omega_i) p(\mathbf{X}_{k-1,n}^a | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d, \omega_i)}{p(\mathbf{z}_k^f | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d, \omega_i)} \quad (3.5)$$

where the denominator term is given in the usual fashion by summing the numerator expression over all  $\mathbf{X}_{k,n}^a$ :

$$p(\mathbf{z}_k^f | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d, \omega_i) = \sum_{\mathbf{x}_{k,n}^a, n=1}^{N_p} p(\mathbf{z}_k^f | \mathbf{x}_{k,n}^a, \omega_i) p(\mathbf{x}_{k,n}^a | \mathbf{x}_{k-1,n}^a, \mathbf{Z}_m^d, \omega_i) p(\mathbf{X}_{k-1,n}^a | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d, \omega_i) \quad (3.6)$$

Each element in the original Larson and Peschon equations has a counterpart in this object recognition application. Thus for any given object model  $\omega_i$ , we can conceptually use the Larson and Peschon approach to find the path  $\mathbf{X}_{k,n}^a$  which maximizes  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ . Note that, whereas the Larson and Peschon approach started with an *a priori* probability density  $p(\mathbf{x}_0^a)$ , the procedure described here would start with an *a priori* probability  $p(\mathbf{x}_{0,n}^a | \mathbf{Z}_m^d, \omega_i)$ .

This process is illustrated in Figs. 3.3 and 3.4, using an abbreviated notation for clarity. Note that in these figures, allowable starting states are restricted to an initial window in the aspect angle region, and allowable transitions are restricted to fall from one window to the next. This reflects an *a priori* judgment that other possibilities are of negligible probability. Fig. 3.3 shows how transitions from window to window for one path (arbitrarily selected from the two sample paths shown) might appear as projected onto the entire

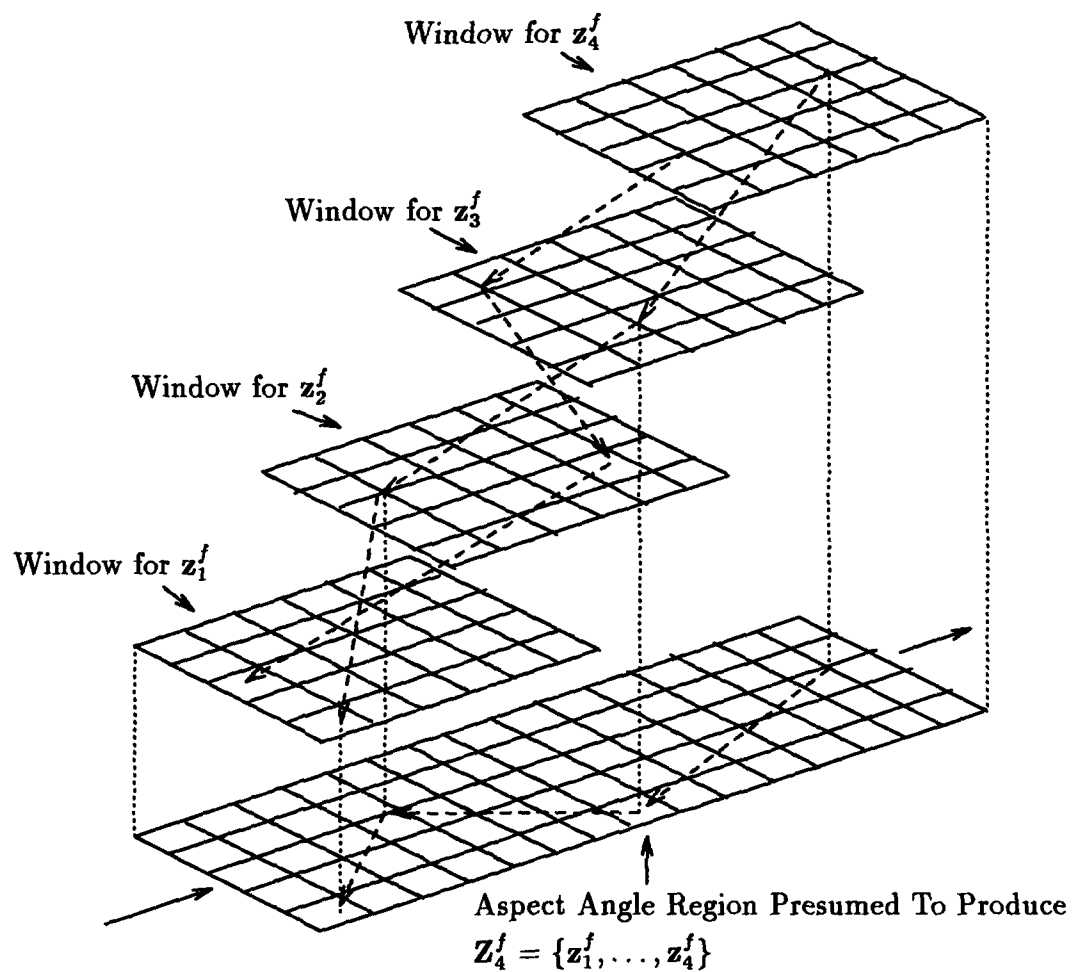


Figure 3.3. Typical Matching Paths - Larson and Peschon Algorithm

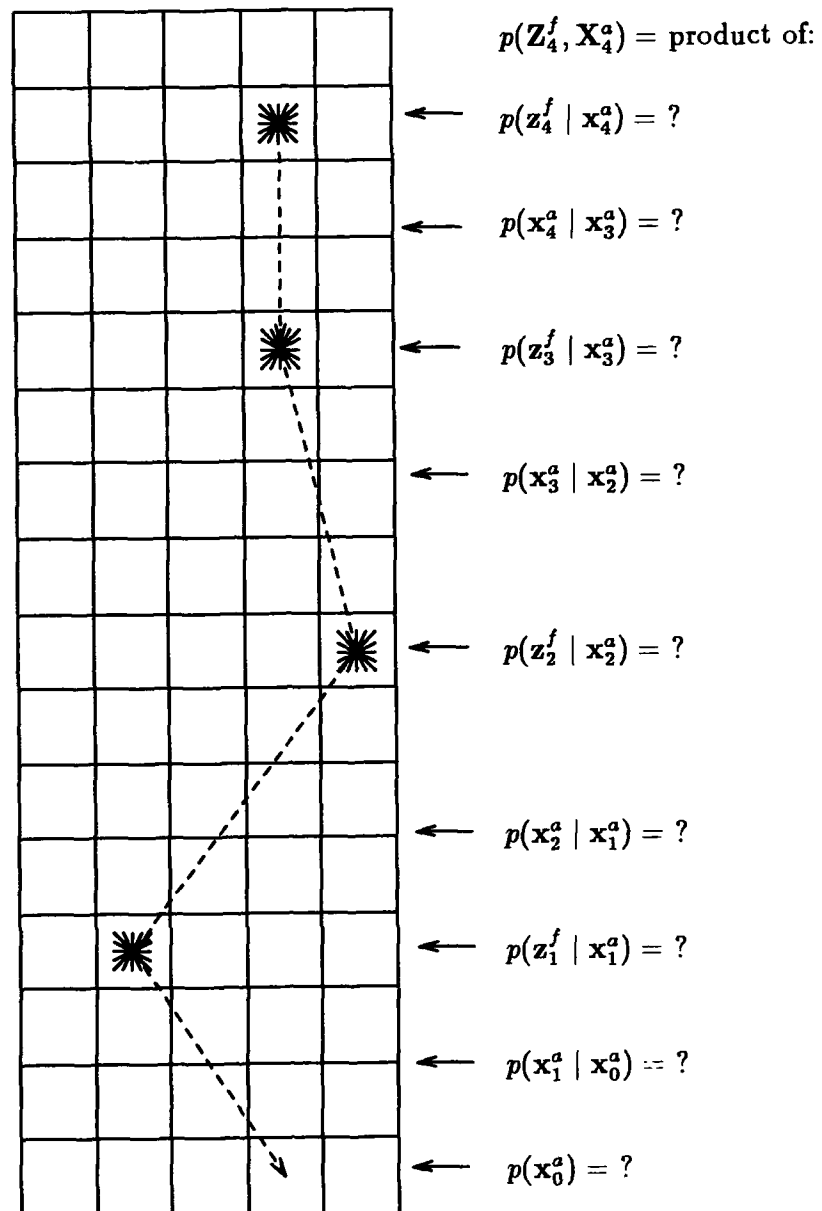


Figure 3.4. "Overhead View" - Larson and Peschon Path



allowable aspect angle region. Note that the joint probability density " $p(\mathbf{Z}_4^f, \mathbf{X}_4^a)$ " found in Fig. 3.4 is simply a shorthand notation (i.e., not showing all appropriate conditioning variables) for the product of the numerator terms in Eqns. (3.4) or (3.5), when  $k = 4$ .

So far, the discussion has been concerned with defining Larson and Peschon-like conditional probabilities for the aspect angle space corresponding to *one* object class  $\omega_i$ . Now, we will modify the Larson and Peschon equations to allow us to consider the *a priori* probability of class membership  $p(\omega_i)$ , and ultimately to define the *a posteriori* probabilities  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  that we seek. Assuming (with reservations as discussed above) that  $p(\omega_i) = p(\omega_i | \mathbf{Z}_m^d)$ , we start with the *a priori* probability  $p(\omega_i)$  and then multiply by  $p(\mathbf{x}_{0,n}^a | \mathbf{Z}_m^d, \omega_i)$ , and continue as in the Larson and Peschon development, to obtain in an analogous fashion:

$$p(\mathbf{X}_{k,n}^a, \omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d) = \frac{p(\mathbf{z}_k^f | \mathbf{x}_{k,n}^a, \omega_i) p(\mathbf{x}_{k,n}^a | \mathbf{x}_{k-1,n}^a, \mathbf{Z}_m^d, \omega_i) p(\mathbf{X}_{k-1,n}^a, \omega_i | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d)}{p(\mathbf{z}_k^f | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d)} \quad (3.7)$$

where the denominator term is:

$$p(\mathbf{z}_k^f | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d) = \sum_{\omega_j, j=1}^J \sum_{\mathbf{X}_{k,n}^a, n=1}^{N_p} [p(\mathbf{z}_k^f | \mathbf{x}_{k,n}^a, \omega_j) p(\mathbf{x}_{k,n}^a | \mathbf{x}_{k-1,n}^a, \mathbf{Z}_m^d, \omega_j) p(\mathbf{X}_{k-1,n}^a, \omega_j | \mathbf{Z}_{k-1}^f, \mathbf{Z}_m^d)] \quad (3.8)$$

Now sum Eqn. (3.7) over all possible  $\mathbf{X}_{k,n}^a$  for any given  $\omega_i$  to obtain:

$$p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d) = \sum_{\mathbf{X}_{k,n}^a, n=1}^{N_p} p(\mathbf{X}_{k,n}^a, \omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d) \quad (3.9)$$

Thus, the desired  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  can be found rigorously only by keeping track of, and performing appropriate calculations for, all possible aspect angle paths over all possible object models – that is, all  $\mathbf{X}_{k,n}^a$  in  $\mathcal{X}_{pk}^a$  over all  $\omega_i$ . In the following section, we will repeat these calculations using an approach closer to the actual algorithm used by Larson and

Peschon, and then we will consider modifications that might be desired to provide more rigorous answers, if the basic assumptions of this development do not apply.

*3.6.3 Relating  $I^*(\mathbf{x}_k^a, k)$  to Object Class Probability  $p(\omega_i \mid \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ .* Recall that the "practical version" of the Larson and Peschon method as implemented on different object models  $\omega_i$  would find the  $\mathbf{X}_{k,n}^a$ , say  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ , for each  $\omega_i$ , that maximizes  $p(\mathbf{X}_{k,n}^a \mid \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ , not by maximizing this conditional probability directly, but rather by maximizing  $I^*(\mathbf{x}_k^a, k)$ , where (using the original Larson and Peschon form, but recalling that in *our* development for object recognition, all probabilities would be conditioned also on  $\mathbf{Z}_m^d$  and  $\omega_i$ ):

$$\text{MAX}_{\mathbf{x}_k^a} I^*(\mathbf{x}_k^a, k) = \text{MAX}_{\mathbf{x}_k^a} \left[ p(\mathbf{z}_k^f \mid \mathbf{x}_k^a) p(\mathbf{x}_k^a \mid \mathbf{x}_{k-1}^a) I^*(\mathbf{x}_{k-1}^a, k-1) \right] \quad (3.10)$$

or, in our form:

$$\begin{aligned} \text{MAX}_{\mathbf{x}_{k,n}^a} I^*(\mathbf{x}_{k,n}^a, k \mid \omega_i) = \\ \text{MAX}_{\mathbf{x}_{k,n}^a} \left[ p(\mathbf{z}_k^f \mid \mathbf{x}_{k,n}^a, \omega_i) p(\mathbf{x}_{k,n}^a \mid \mathbf{x}_{k-1,n}^a, \mathbf{Z}_m^d, \omega_i) I^*(\mathbf{x}_{k-1,n}^a, k-1 \mid \omega_i) \right] \end{aligned} \quad (3.11)$$

Maximizing this quantity rather than the conditional probability is desirable because we avoid having to compute values for all  $\mathbf{X}_{k,n}^a \in \mathcal{X}_{pk}^a$ , which we would have to do to find the denominator term as in Eqn. (3.6). Examining  $I^*(\mathbf{x}_{k,n}^a, k \mid \omega_i)$  closely, note that the preceding equation is equivalent to:

$$\text{MAX}_{\mathbf{x}_{k,n}^a} I^*(\mathbf{x}_{k,n}^a, k \mid \omega_i) = \text{MAX}_{\mathbf{X}_{k,n}^a} \left[ p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f \mid \mathbf{Z}_m^d, \omega_i) \right] \quad (3.12)$$

Thus, the practical version of the Larson and Peschon equations gives the maximum value of  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f \mid \mathbf{Z}_m^d, \omega_i)$ , and the state history estimate  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$  for a given  $\omega_i$  which gives that maximum joint conditional probability density, or, equivalently, since there is only one  $\mathbf{Z}_k^f$ , that state history which maximizes  $p(\mathbf{X}_{k,n}^a \mid \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ .

Suppose on the other hand that we had chosen to find  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f, \omega_i \mid \mathbf{Z}_m^d)$  for all  $\mathbf{X}_{k,n}^a \in \mathcal{X}_{pk}^a$ . This can be had by computing  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f \mid \mathbf{Z}_m^d, \omega_i)$  for each  $\mathbf{X}_{k,n}^a$  over each  $\omega_i$

and multiplying each such quantity by the appropriate  $p(\omega_i)$  (again, as above, taken equal to  $p(\omega_i | \mathbf{Z}_m^d)$ , with the same qualifications) to find:

$$p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f | \mathbf{Z}_m^d, \omega_i) p(\omega_i) = p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f, \omega_i | \mathbf{Z}_m^d) \quad (3.13)$$

But:

$$p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d) = \sum_{\mathbf{X}_{k,n}^a, n=1}^{N_p} p(\mathbf{X}_{k,n}^a, \omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d) \quad (3.14)$$

or:

$$p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d) = \sum_{\mathbf{X}_{k,n}^a, n=1}^{N_p} \left[ \frac{p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f, \omega_i | \mathbf{Z}_m^d)}{p(\mathbf{Z}_k^f | \mathbf{Z}_m^d)} \right] \quad (3.15)$$

where:

$$p(\mathbf{Z}_k^f | \mathbf{Z}_m^d) = \sum_{\omega_j, j=1}^J \sum_{\mathbf{X}_{k,n}^a, n=1}^{N_p} p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f, \omega_j | \mathbf{Z}_m^d) \quad (3.16)$$

or:

$$p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d) = \frac{\sum_{\mathbf{X}_{k,n}^a, n=1}^{N_p} p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f, \omega_i | \mathbf{Z}_m^d)}{\sum_{\omega_j, j=1}^J \sum_{\mathbf{X}_{k,n}^a, n=1}^{N_p} p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f, \omega_j | \mathbf{Z}_m^d)} \quad (3.17)$$

which is equivalent to the expression in Eqn. (3.9), requiring consideration of probabilities for all paths  $\mathbf{X}_{k,n}^a$ , over all object models  $\omega_i$  – an exhaustive search.

In identifying the fact that  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  can be found properly only by exhaustive search, we have reached the fundamental objective of this Bayesian development. The assumptions that were made in pursuing this objective can be relaxed without changing this fundamental result, by implementing the suggestions made above where those assumptions were introduced, and repeating the development. Summarizing key issues, those relaxed assumptions could include: (1) allowing for an infinite number of aspect angle locations (a continuous space); (2) allowing for the generation of a single feature observable measure-

ment from a set of aspect angle locations, rather than one; (3) allowing for inclusion of *a priori* information on maneuver likelihoods for different classes, and (4) obtaining maximum *a posteriori* probabilities, rather than maximum likelihoods, for signature-to-aspect angle associations and other events where uncertainty exists in the origin conditions for the observed signature (due to uncertain range registration, for example).

**3.6.4 Approximations for  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ , and Associated Problems.** In any case, it is clear that the *a posteriori* probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  will not be practical to obtain – attempting to keep track of conditional probabilities for all tracks  $\mathbf{X}_{k,n}^a$  over all object models  $\omega_i$  would be in general a monumental task, to be avoided if at all possible. Modifying the right side of Eqn. (3.17) to take summations over prescribed sets of  $\mathbf{X}_{k,n}^a$ 's rather than all (e.g., *proper subsets* of  $\mathcal{X}_{pk}^a$ ) for each  $\omega_i$  in Eqn. (3.17) creates a limiting process, so that as we converge toward summations over all  $\mathbf{X}_{k,n}^a \in \mathcal{X}_{pk}^a$  for each  $\omega_i$ , the modified term converges toward the desired probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ .

We recognize that most of the paths will contribute little to the final probability calculations. Clearly, the path which contributes the most to the probability calculation for each  $\omega_i$  is the one given by the Larson and Peschon equations,  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ . Therefore, if we make the (extreme) choice of approximating the desired  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  using only one path  $\mathbf{X}_{k,n}^a$  for each  $\omega_i$ , the most reasonable such approximation would be given by (note the "hat" over  $p$ , denoting an estimate):

$$\hat{p}(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d) = \frac{p(\hat{\mathbf{X}}_{k/\omega_i}^{LP}, \mathbf{Z}_k^f, \omega_i | \mathbf{Z}_m^d)}{\sum_{j=1}^J p(\hat{\mathbf{X}}_{k/\omega_j}^{LP}, \mathbf{Z}_k^f, \omega_j | \mathbf{Z}_m^d)} \quad (3.18)$$

where, using the appropriate  $I^*(\mathbf{x}_{k,n}^a, k | \omega_i)$  for each  $\omega_i$ :

$$p(\hat{\mathbf{X}}_{k/\omega_i}^{LP}, \mathbf{Z}_k^f, \omega_i | \mathbf{Z}_m^d) = \left[ \max_{\mathbf{x}_{k,n}^a} I^*(\mathbf{x}_{k,n}^a, k | \omega_i) \right] p(\omega_i) \quad (3.19)$$

We assume that the aspect angle spaces for the various object models are adequately discretized, or sampled in a Shannon sampling sense, so that variations in feature observables from one discrete aspect angle state to the next are "small" in some sense – for

example, an expected feature observable value from a (continuous) aspect angle value between two adjacent discrete aspect angle values should be well approximated as a linear interpolation of the feature observable values for the two adjacent discrete states.

Then, progressively better approximations to  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  would be given by summing also the contributions from paths for each  $\omega_i$  that pass through points  $\mathbf{X}_{k,n}^a$  in progressively larger neighborhoods around the points in  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$  for each  $\omega_i$ . An alternative and perhaps better approach would be to consider some number of additional paths  $\mathbf{X}_{k,n}^a$  (for fair comparison, the same number of additional paths for each class  $\omega_i$ ) tracing back from terminal points  $\mathbf{x}_{k,n}^a$  that resulted in values of  $I^*(\mathbf{x}_{k,n}^a, k | \omega_i)$  closest to the maximum value which defined  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ . A third alternative would be to consider contributions from aspect angle states that lie in between the elements of  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ , but which are skipped along that path, or not included in  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$  *per se*. Consideration of this third alternative will lead back to classical dynamic programming-based sequence comparison methods in Sect. 3.7.

In summary, we have obtained in this section an expression for the probability of object class membership via Bayesian techniques, where the *new information* (in a Bayesian sense) is given by feature space measurements, conditioned on the likelihood of required kinematic transitions in the underlying aspect angle state space. This conditioning provides the restriction of feature space matches to kinematically likely subsets, as discussed in Sect. 2.6. Due to the assumed highly nonlinear and complex nature of the relationship between transitions in aspect angle and feature space, we used a Larson and Peschon estimation approach, which in general forces us to approximate by using only a finite number (perhaps one) of likely aspect angle state sets (sequences) for each object class.

**3.6.5 Obtaining "A Priori" Information for the L&P Approach.** A significant question left unanswered in the previous section is how to obtain the *a priori* knowledge of aspect angle transitions, expressed as the aspect angle state (cell) transition probability or likelihood  $p(\mathbf{x}_{k+1,n}^a | \mathbf{x}_{k,n}^a, \mathbf{Z}_m^d, \omega_i)$  and probability of starting cell location  $p(\mathbf{x}_{0,n}^a | \mathbf{Z}_m^d, \omega_i)$ . In object recognition, this is simply information that we have about the object-sensor angular position and rate, apart from what we may learn from the latest feature observable measurement. If our only other source of object information is a conventional "kinematic"

object tracker using prior and current position, (pointing) angle, and range rate measurements against highly dynamic objects like aircraft capable of sudden turns, we may rightly feel that information on angular position and rate from this source is of little value. At least two reasonable cases for defining  $p(\mathbf{x}_{j+1,n}^a \mid \mathbf{x}_{j,n}^a, \mathbf{Z}_m^d, \omega_i)$  from kinematics exist, however.

First, if we are willing to accept non-real time estimates, we can smooth our kinematic measurements to provide a high-quality estimate of object velocity and acceleration. If the object is an aircraft determined to be in a turn, with reasonably constant acceleration relative to the target body frame for a period of several seconds, a smoother can be made to yield an estimate of target state sufficiently adequate so as to allow a reasonable determination of aspect angle from kinematic (smoother) information alone. A L&P estimator using this concept is discussed in Chapter V.

Second, recall that the Kendrick-type filter provides a high-quality velocity and acceleration estimate for aircraft targets by considering the target aerodynamic state implied by aspect angle measurements. This implies a reasonable real-time estimate for  $p(\mathbf{x}_{j+1,n}^a \mid \mathbf{x}_{j,n}^a, \mathbf{Z}_m^d, \omega_i)$  from a Kendrick-type filter. Conversely, however, the Kendrick-type filter requires a reasonable aspect angle estimate, which may not be available from conventional pose estimators in some feature spaces. Such an estimate is, however, available from the L&P-type estimator, as the latest state  $\hat{\mathbf{x}}_k^a$  in the sequence  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ . We will return to this thought in Sect. 3.9.

Note that each of these two cases prescribes only a *minimum-sufficient* set of conditions for providing reasonable *a priori* aspect angle transition information on aircraft targets to a Larson and Peschon-type algorithm. In the first case, we are able to use a simple tracking filter and kinematic measurements only, but surrender real-time operation by the need to smooth. In the second case, we accept a more complicated tracking filter, but recoup the possibility of real time information. If, on the other hand, we desire to have the best possible information quality regarding aspect angle transition, the proper answer is perhaps some form of integrated kinematic/aspect-angle filter *and* a smoother. Simply, we should always be able to estimate better with more information than we can with less.

In any case, however, we see that the time period over which kinematic measurements  $Z_m^d$  are taken must encompass that over which feature observable measurements  $Z_k^f$  are taken. In the first case, the smoothing process will require that kinematic measurements precede and follow the time interval over which feature observables are measured. In the second case, it will be desirable that initial kinematic measurements precede feature observable measurements, so that target velocity information is available for use with initial feature observable measurements. This latter case was true in Sect. 3.4.2, for which we effectively had  $m = k + s$  with  $s \geq 1$ .

A final caution is in order regarding the estimation of aspect angle states or transitions from translational kinematics. The quality of these estimates is driven by (1) the extent to which aspect angle and kinematics are linked for each object class, and (2) the extent to which we properly *understand* that linkage for each possible object class. For ground vehicles for example, a fundamental problem may arise from assumptions that a vehicle is turning on a flat surface. For helicopter targets, aspect angle and translational kinematics are very poorly linked in some flight regimes – in such cases, we may be able to say only that aspect angle change with time is bounded, although with unpredictable direction.

With respect to aircraft, the reader familiar with developments in control-configured flight will recognize that many of our concerns about the ability to predict aspect angle from kinematic measurements arise from the new possibilities presented by these developments, relative to conventional “coordinated turn” flight. An interesting discussion of these possibilities and underlying tactical requirements is found in [112]. Some of the new capabilities include high angle of attack flight for maneuverability, pitch pointing for gun aiming without change of flight path angle, zero angle of attack turns that keep the wind vector aligned with the body  $\vec{x}_b$  or longitudinal axis, and so on.

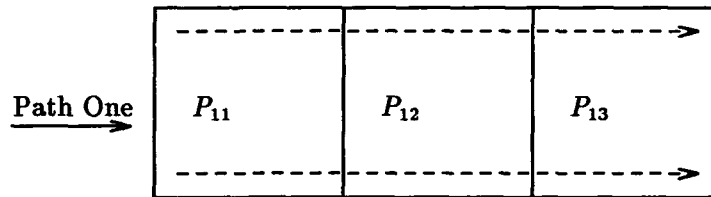
All of these possibilities complicate the problem, but do not make recognition impossible by any means. If target behavior *hypotheses* in addition to the coordinated turn model are feasible and expected for one or more target classes, the recognizer can be designed to consider them. This increases the dimensionality of our problem, which is already naturally expressed as a multiple model (object or target class) estimator – the fundamental approach discussed in Sect. 2.3.1.3, whether or not the models are implemented as Kalman

filters. However, as long as at least one of our hypotheses is close in some sense to the actual target behavior, correct recognition remains feasible. In fact, the computation of aspect angle estimates from translational kinematics is simpler for some of these options (e.g., the zero angle of attack turns noted above) than for the coordinated turn hypothesis assumed for aircraft targets here. Thus, implementation of models for these hypotheses in a multiple model recognizer or estimator will be more straightforward than implementation of the model for the fundamental coordinated turn hypothesis. In nearly every case, however, target kinematics provides *some* information about aspect angle – our intent is to use that information.

**3.6.6 Issues in the Use of Larson and Peschon Methods in Object Recognition.** As implied in Sect. 2.4.5 of the last chapter, there is a potential pitfall of a “basic” Larson and Peschon approach as applied to object recognition (e.g., Eqn. (3.18)) – making decisions based on but one set of  $k$  aspect angle states or one path per object model. In fact, assuming equal *a priori* probabilities for each  $\omega_i$ , the path which yields the highest  $I^*(\mathbf{x}_{k,n}^a, k | \omega_i)$  over all  $\omega_i$  may *not* fall on the particular  $\omega_i$  which has the highest actual *a posteriori* probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  of being the correct class. It *may* be that one model has a particularly “well-configured” set of aspect angle states and associated feature observable values, such that the “best” path  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$  on this model traverses these states, and give this model the highest  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f | \mathbf{Z}_m^d, \omega_i)$ , but if all *possible*  $\mathbf{X}_{k,n}^a$  are considered, this model is less likely to have been the origin of the observed  $\mathbf{Z}_k^f$  than some *other* model.

An example is helpful to illustrate this issue (see Fig. 3.5). Suppose we have two contiguous sequences of three aspect angle “pixels” each. Presume that we know that the object presented one of the two sequences to the sensor at some imprecisely known rate over some interval of time, including the time  $t_m$  at which a measurement  $z$  was received. The probability density or likelihood that  $p(z | \mathbf{x})$  that state  $\mathbf{x}$  would yield the (scalar) measurement  $z$  is taken to be Gaussian, with mean and variance shown for each pixel. Now suppose the measured value is “ $z = 3.05$ ”. If required to choose between the two paths, a basic L&P-type algorithm would choose path one, since this path contains one pixel,  $P_{12}$ , most likely to have produced the given observation.

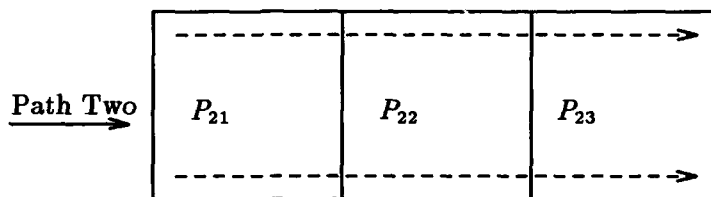




$p(z | x = P_{11})$ : Gaussian with mean 6.0, standard deviation 1.0

$p(z | x = P_{12})$ : Gaussian with mean 3.0, standard deviation 1.0

$p(z | x = P_{13})$ : Gaussian with mean 6.0, standard deviation 1.0



$p(z | x = P_{21})$ : Gaussian with mean 2.95, standard deviation 1.0

$p(z | x = P_{22})$ : Gaussian with mean 2.95, standard deviation 1.0

$p(z | x = P_{23})$ : Gaussian with mean 2.95, standard deviation 1.0

Figure 3.5. Two Aspect Angle Paths

But, given our imprecise information on the transition of the sensor over these pixels, it is perhaps almost equally likely that the first or third cells could have generated the received measurement. Which path is truly more likely to have generated the observation? Using the full Bayesian approach as in Eqn. (3.9) or (3.17), we would select path two.

Now observe that a classical sequence comparison or “dynamic time warping” concept with simple continuity rules, unable to delete pixels and “skip” pixels  $P_{11}$  and  $P_{13}$ , would have picked path two, since, by any reasonable metric, this path is more likely *over its extent* to have produced the observed  $z$ .

This example illustrates a potential weakness in L&P-based concepts for object recognition by aspect angle path determination, and indeed, a caution in the use of the Larson and Peschon method. Recall that in the Barniv and Kramer applications of the Larson and Peschon method, we had visibility over the individual states, i.e., cells or sequences

of cells overlying some physical space. The origin state of a given measurement was never ambiguous – the ambiguity lay in whether or not a given string of measurements over some states represented a real target. *This is not so in object recognition* when we take, as states, “cells” of aspect angle on some hypothetical aspect angle sphere – here we never really *know* which aspect angle state (cell) generated which measurement – only that a particular measurement *was* generated as a certain set of states was passed through or over.

Thus, for the object recognition application, when the number of cells or aspect angle states outnumber the number of measurements, we are perhaps better advised to ask the question, “*given that we have traversed a continuous path over one of several hypothetical aspect angle spheres, which sphere is more likely to have yielded the observed sequence of discrete observations?*” The issues here are precisely those discussed in relation to Fig. 3.1: what is the relationship between the feature observables, state space, and likely transitions on that state space as measurements occur?

One approach to working this problem is to consider contributions from multiple trajectories as discussed in Sect. 3.6.4. An alternative approach to using the L&P format would be classical sequence comparison or dynamic time warping-based methods. First, one would construct sets of trajectories through the state space over the time frame of interest, using the same information on aspect angle from kinematics used to provide  $p(\mathbf{x}_{0,n}^a \mid \mathbf{Z}_m^d, \omega_i)$  and  $p(\mathbf{x}_{j+1,n} \mid \mathbf{x}_{j,n}, \mathbf{Z}_m^d, \omega_i)$ , the *a priori* information for L&P-type approaches. These trajectories then imply sequences of feature observations, which can be compared to the observed sequences using classical sequence comparison or DTW-like techniques. Further, in a departure from usual DTW, we can allow the “best path” to move from one trajectory to another. This defines a “two-dimensional” form of classical sequence comparison, as in Fig. 2.9.

Users of Larson and Peschon-like approaches for object recognition such as those to be demonstrated in Chapter V must keep these issues in mind. Certainly, one may conduct studies in a straightforward Monte-Carlo manner to determine the probabilities of false or correct class assignment when using an approximate approach like that in Eqn. (3.18), as compared to a more probabilistically correct approach like that given in Eqn. (3.17). It should also be clear that all such expressions for  $p(\omega_i \mid \mathbf{Z}_k^f, \mathbf{Z}_m^d)$  are in fact *generalized*

*likelihood functions* and, as such, are suitable for analysis using the techniques discussed in Sect. 2.7, to be further discussed in Sect. 3.11.

### 3.7 Classical Sequence Comparison in Object Recognition.

In the last several sections, we developed the Larson and Peschon equations for object recognition. While reviewing some potential problems with Larson and Peschon-like approaches to sequence comparison, we observed that classical sequence comparison might serve to overcome those problems. In this section, we will construct object recognition as a problem in classical sequence comparison, and consider some problems inherent in this approach. Ultimately, the lesson is that the Larson and Peschon equations and classical sequence comparison are intimately related, and should be considered as “tools” for solving sequence comparison problems (as indeed may other algorithms, like hidden Markov models [176], etc.) – particular tools may serve particular scenarios better than others.

All of these sequence comparison or “motion warping” tools will serve to a greater or lesser degree to achieve our goal – restricting the matching domain of object recognition algorithms to kinematically likely sets. Thus, the “likelihood function” outputs of these algorithms will reflect, to a greater degree than for conventional “independent look” algorithms, the *joint likelihood* of observed features (signatures) and kinematics.

**3.7.1 Motivating Classical Sequence Comparison in Object Recognition.** In Sect. 3.6, we constructed object recognition as a problem in Larson and Peschon estimation. In this section, we will view the same problem from a classical sequence comparison perspective. Figure 3.6 illustrates a baseline case of sequence comparison for multiple models. The figure shows three object models, or hypothetical aspect angle spheres, for which discrete signatures are recorded *a priori*. A sequence of true signatures or feature observables is extracted from the topmost or true object, which is of course not identified *a priori* to the recognizer. Also, derived generally in the manner by which the Kendrick estimator in Sect. 2.3.3 found the kinematically-implied aspect angle “pseudo-measurement,” we have a kinematically-implied aspect path for each object model.

Fig. 3.6.a

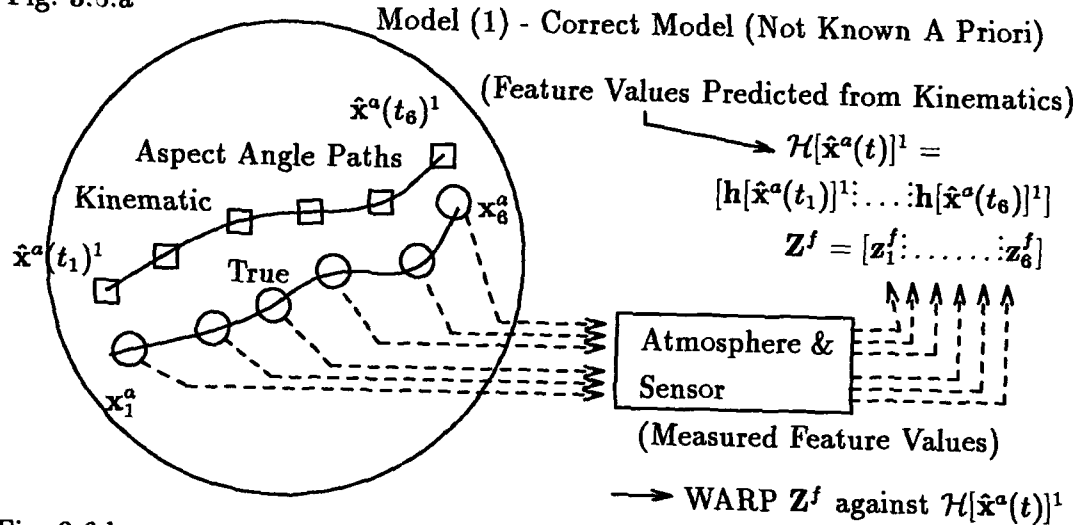


Fig. 3.6.b

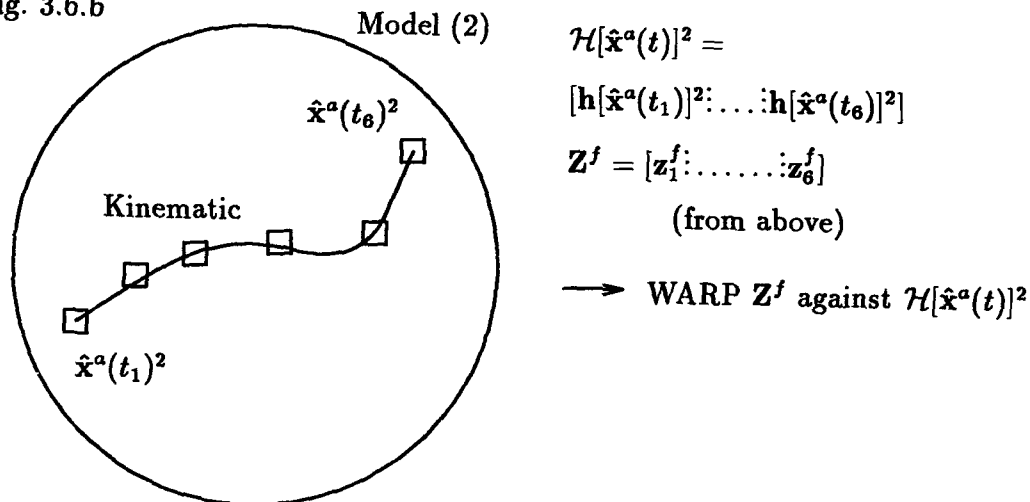


Fig. 3.6.c

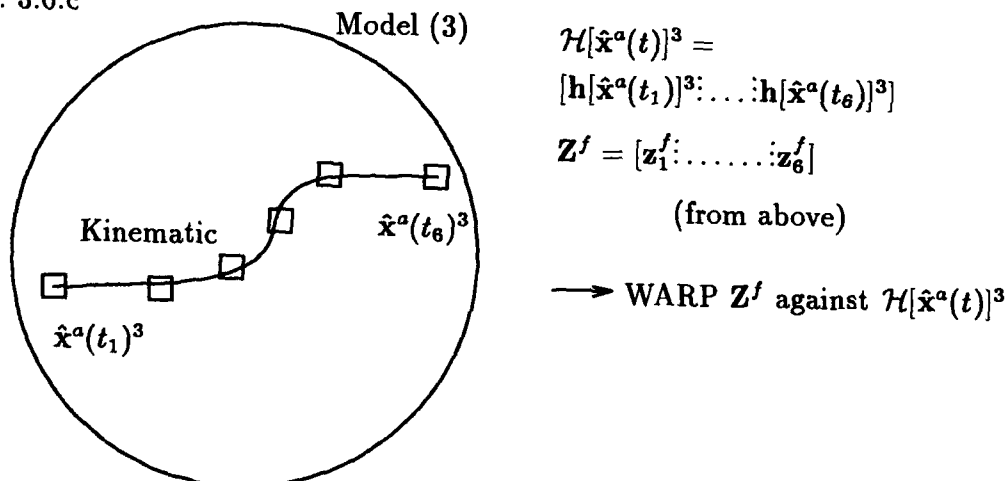


Figure 3.6. Motion Warping - Aspect Angle Paths and Multiple Models

Restating the problem using the variable conventions of Sect. 3.6, we have a sequence of measured feature observable values  $\mathbf{z}^f$  extracted over time to form the measurement history  $\mathbf{Z}^f$ . Each element in the history corresponds to an element in the sequence, i.e., an observed value, function (e.g., a radar range sweep measurement) or again a vector of values at a particular measurement time. The kinematically-implied paths or aspect angle state sequences are the same ones that would be found by propagating maximum likelihood values of the Larson and Peschon *a priori* information  $p(\mathbf{x}_{j+1,n}^a | \mathbf{x}_{j,n}^a, \mathbf{Z}_m^d, \omega_i)$  and probability of starting cell location  $p(\mathbf{x}_{0,n}^a | \mathbf{Z}_m^d, \omega_i)$  for each object model throughout the entire time interval of interest.

Along that "kinematic path", from each model  $j$  of  $J$  object models, we can extract a sequence (vector)  $\hat{\mathbf{Z}}^f = \mathcal{H}[\hat{\mathbf{x}}^a(t)]$  of estimated or predicted feature observable values  $\hat{\mathbf{z}}_i^f = \mathbf{h}[\hat{\mathbf{x}}^a(t_i)]$  for times  $t_i$  corresponding to the times at which the elements of  $\mathbf{Z}^f$  were observed. The labeling convention used here is as in the discussion on tracking filters previously, but the calligraphic letter "H" or  $\mathcal{H}$  distinguishes the sequence of predicted measurements from the measurement *matrix*  $\mathbf{H}$  discussed in Sect. 2.3 in connection with the Kalman filter.

Note that this procedure will make no attempt to use *individual* measured feature observable values to determine "pose estimates" for a maximum likelihood "feature observable-based aspect path" on each model, as in Fig. 2.6. Thus, the models show no such path.

If (1) the model class  $\omega_i$  corresponds exactly to the object class, (2) the object is moving with the kinematics assumed for the model, (3) the object signature is non-random, and (4) if our atmospheric transmission and sensors are noise-free, then  $\mathbf{Z}^f$  and  $\mathcal{H}[\hat{\mathbf{x}}^a(t)]$  will be equal. This situation is illustrated in Fig. 3.7, in which  $\hat{\mathbf{x}}^a(t_k)$  represents the  $k$ th point in the sequence of kinematically-estimated aspect angles, while  $\mathbf{x}^a(t_k)$  represents the  $k$ th point in the sequence of true aspect angles. Note for future consideration that here the kinematically-estimated and true paths trace identical aspect angles at identical times.

In general, however, these conditions will not be satisfied: (1) there will be only one observed sequence  $\mathbf{Z}^f$ , but there will be one anticipated sequence  $\mathcal{H}[\hat{\mathbf{x}}^a(t)]$  for *each*

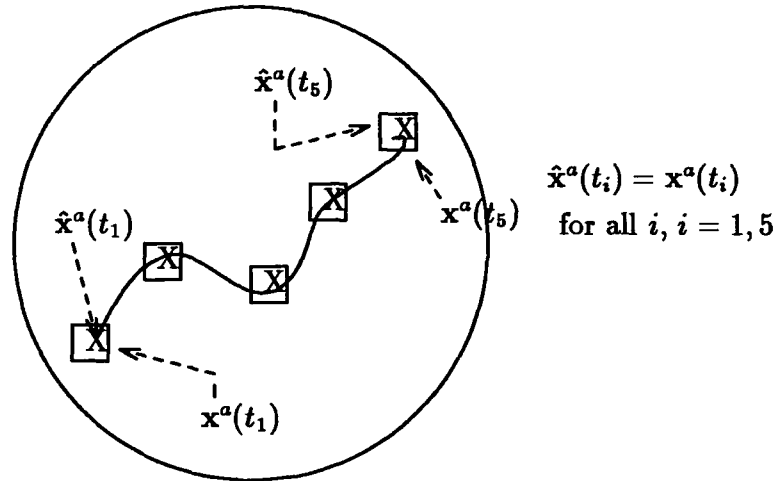


Figure 3.7. Complete Correspondence Between Kinematically Estimated and True Aspect Angle Paths

possible object (model) class, as we saw in Fig. 3.6 (the origin object class of  $Z^f$  is, of course, not known *a priori*); (2) the object kinematics will not be perfectly known; (3) the object signature generation is a random process, even for constant aspect angle, and (4) atmospheric transmission and sensors add further random noise to the signature.

Note in Fig. 3.6 that the kinematic aspect path and the true path can properly be shown together only on one model – the true (unknown *a priori*) model. This is because the *true* path is generated by only *one* model (class), not all. If the true and kinematic aspect paths for this true model lie on the same aspect angle path, but traverse discrete signature origin points along that path at different times, then, in the absence of other random factors, the measured feature observable values and the kinematically-estimated feature observable values for the correct object model will be equal except for sections of relative compression and expansion. These differences between the resulting sequences can be resolved with the usual “one-dimensional” classical dynamic programming sequence comparison, or “warping” process discussed in Sect. 2.4.2.

The warping process takes place between the elements of  $Z^f$  and each  $\mathcal{H}[\hat{x}^a(t)]$  – the pairing of  $Z^f$  and some  $\mathcal{H}[\hat{x}^a(t)]$  for which the dynamic programming-based sequence comparison cost is least is taken to indicate the correct class  $\omega_i$  for the object which yielded  $Z^f$ . We may note also that, in general, the two sequences to be compared need not have an

equal number of elements. Dynamic programming-based sequence comparison techniques handle small mismatches in the total number of elements naturally by "expanding" the shorter sequence as required to achieve a best comparison.

Where the true path and kinematic path lie side by side, or cross, we will call this a condition of "off-nominal path" errors, where the nominal path is in fact the kinematically-estimated path. Off-nominal path errors are expected to be due to errors in modeling or estimation of aspect angle from kinematics, as discussed in Sect. 3.6.5. Consider Fig. 3.6.a as representing a case of a nominal aspect angle path and the true path separated by an off-nominal error. Off-nominal path errors cannot be accounted for by "one-dimensional" classical sequence comparison or "warping", but if the off-nominal path error is reasonably small and the feature observables are reasonably constant in directions normal to the two paths, it may still be possible in general to remove some portion of the errors due to compression and expansion, and perhaps to identify the best object-model match over the whole trajectory.

It should be clear that most of the illustrations so far in this chapter have shown the general case in which we have off-nominal path errors. If we attempt only a single "one-dimensional" sequence comparison for each model, it is possible to conceive of situations in which off-nominal path errors would lead to misclassification.

As implied in the discussion of Sects. 2.4.3 and 2.4.5, however, extensions to classical sequence comparison can deal directly with off-nominal path errors. Rather than comparing our observed sequence to a single one-dimensional sequence from each object model, we may compare our observed sequence to a number of sequences taken from a two dimensional aspect angle region on each object model, as shown in Fig. 2.9.

*3.7.2 Implementing Classical Sequence Comparison in Object Recognition.* The implementation of one and two-dimensional sequence comparison techniques on a multiple-window aspect angle region is illustrated in Fig. 3.8. This figure uses the visual format common to our earlier discussions.

The dotted line labeled "1-D Track" in Fig. 3.8 shows a "one-dimensional" set of aspect angle states against which classical sequence comparison is attempted. It is very

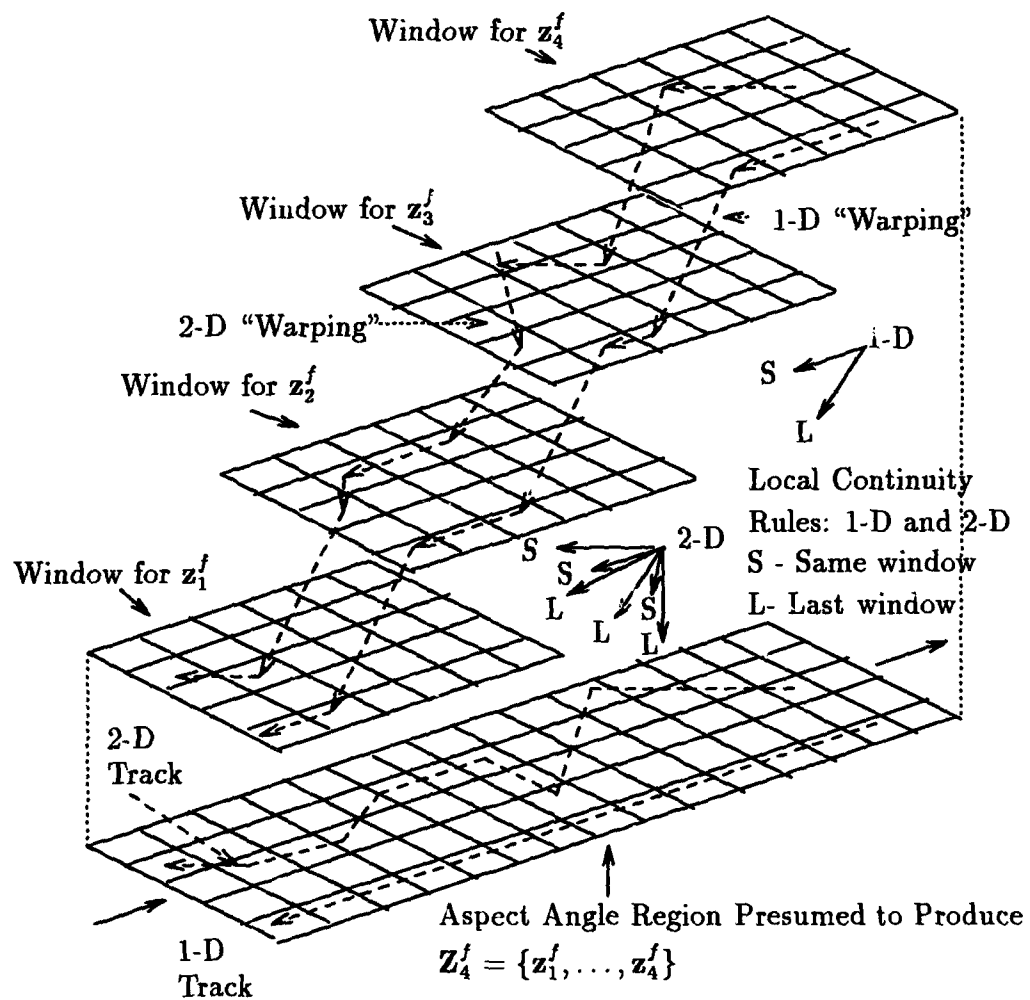


Figure 3.8. Typical Matching Paths – Classical Sequence Comparison



important to note the relationship between Fig. 2.8 and Fig. 3.8 – the first figure is functionally identical to what we would see by viewing Fig. 3.8 from the right side, for any one path. A “window” in Fig. 3.8 is directly analogous to a horizontal layer of circles between the dotted lines in Fig. 2.8.

The continuity constraints for motion warping by one dimensional classical sequence comparison are again directly related to those for dynamic time warping and similar concepts. The global path bounds (dotted lines in Fig. 2.8) are defined by the individual window extents shown in Fig. 3.8. The 1-D local continuity constraints in Fig. 3.8 are exactly those of Fig. 2.8, except that vertical transitions have been forbidden, for reasons to be discussed below. Thus, relative to some “current” point for which costs are to be computed, the predecessor points may lie to the rear on the same window (horizontal transition), or one step to the rear and down on the previous window (diagonal transition).

Note that a horizontal transition corresponds to associating the same measurement with more than one element in the aspect angle region, while a diagonal transition corresponds to a transition from associations for one measurement to associations for the next measurement. The 2-D local continuity constraints in Fig. 3.8 are a simple extension of the 1-D local continuity constraints, allowing for a predecessor point to be located to the right or left of the current point – that is, on a neighboring track parallel to the direction of expected aspect angle change.

The only additional continuity constraints are ones that provide for minimum and maximum numbers of horizontal associations or transitions on each window – that is, a given measurement or observation must be associated with a minimum number of aspect angle cells, but cannot be associated with more than some maximum number of cells. More complicated continuity rules can limit multiple consecutive diagonal transitions, or allow skipping of aspect angle cells (“deletions” from the object signature map, in the sense used by [195]), and so on.

Vertical transitions are forbidden in our case since a vertical transition implies that no aspect angle transition took place between two measurements – two successive measurements are associated with the same aspect angle location. This will be practically

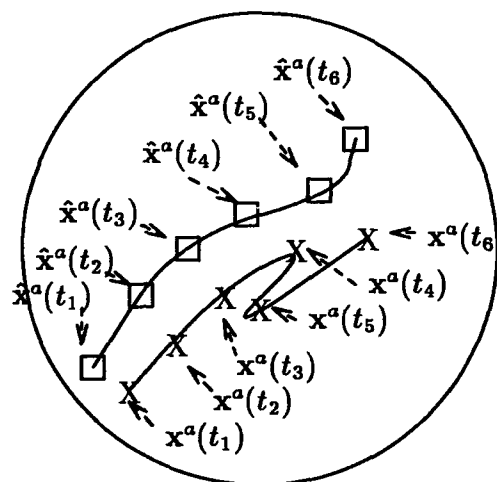
impossible for the scenarios in which this algorithm will be applied, and is therefore excluded from consideration *here*. In some scenarios (e.g., stationary targets), it may be necessary to provide for vertical transitions in the continuity constraints. Also, for cases in which there are more aspect angle cells in the direction of likely transitions than there are measurements to be associated, elimination of vertical transitions helps to prevent undesirably short association paths in the warping path space (associating the measurements with too few aspect angle locations). We will return to the subject of path length compensation in Chapter V.

Resilience to off-nominal path errors was of course a strong point for Larson and Peschon-based approaches defined in Sect. 3.6. These approaches naturally worked on two-dimensional aspect angle regions, and allowed one to quantify the effect of off-nominal path errors, in a Bayesian probabilistic structure.

The differences between sequences expected from kinematic measurements and observed sequences may be more complicated than the simple off-nominal errors shown above. Fig. 3.9 illustrates a case in which the true aspect angle path actually doubles back on itself for a short period. Here, this "wobble" is not picked up by the kinematic sensors and thus is not reflected in the kinematically-derived aspect path.

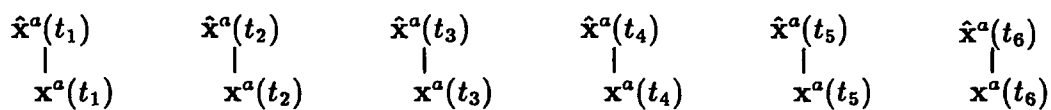
Note that if a linear (non-warped) sequence comparison were made between the kinematic and measured sequences (assume negligible sensor noise on the feature observable measurements) the *effective* aspect angle pairings would be as shown in the first point-to-point association below Fig. 3.9. With dynamic programming-based sequence matching techniques, however, we should be able to determine that a more reasonable match is as shown in the second point-to-point association below Fig. 3.9.

Dealing with cases like the one shown here will require us to be careful in designing those continuity constraints (see Sect. 2.4.2) employed to restrict the dimensionality of dynamic programming decisions. These may prevent us from making certain optimal point-to-point assignments. In the case in Fig. 3.9, for example, we would probably desire to associate point  $\hat{x}_3^a$  with point  $x_5^a$ , but since point  $\hat{x}_4^a$  is associated with point  $x_4^a$ , the former, desired association will be prohibited in classical sequence comparison - this would

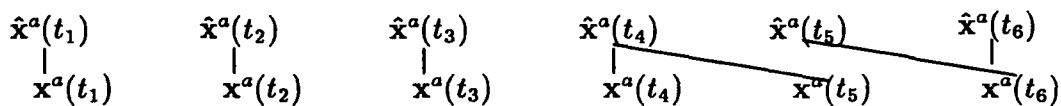


### Kinematic Aspect Angle to True Aspect Angle

Association Without Warping:



Association With Warping:



Match of  $\hat{x}^a(t_3)$  to  $x^a(t_5)$  not permitted here.

Figure 3.9. Kinematic-to-True Aspect Angle Association Improvement With Warping

amount to moving backward in the dynamic programming assignment matrix. "Backward" path options are generally forbidden in classical sequence comparison to limit the *Curse of Dimensionality*.

These classical sequence comparison continuity constraints, or  $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$  in the Larson and Peschon formulation, or other restrictions on aspect angle transitions, can be *tuned* to allow the sequence comparison process to include such folding or other errors in the estimate of aspect angle transition from kinematics. *Any loosening of restrictions must be done carefully, however, because we want the restriction of matching paths to reasonable aspect angle progressions to be the significant factor which increases matching costs for incorrect object-model associations.*

### 3.8 Sequence Comparison Concept Relationships

At this point, we have established four concepts for the use of dynamic programming in dynamic object and target recognition: (1) The Larson and Peschon approach, (2) classical sequence comparison techniques (including dynamic time warping), (3) the Le Chevalier (*et al.*) approach, and (4) the Mieras (*et al.*) approach. This section will illuminate the differences and similarities in these approaches.

Fig. 3.10 is provided as an aid for this discussion. It is intended to show relationships or the lack thereof between concepts that relate to the author's original research, which is noted in the boxed region. Fundamentally, the author's research blends concepts from two fields – linear (and linearization-based) estimation and dynamic programming. The author's conceptualization of dynamic programming as a tool for moving object recognition was preceded *independently* by, but includes and transcends, the developments of Le Chevalier and Mieras – to represent them specifically as applications of dynamic programming sequence comparison for automatic object recognition (AOR) of *dynamic* objects, these names are contained in double boxes.

The key point expressed in the upper portion of Fig. 3.10 is that dynamic programming sequence comparison can be seen to be applicable to multisensor fusion for dynamic object recognition through at least three different paths. The leftmost, or speech process-

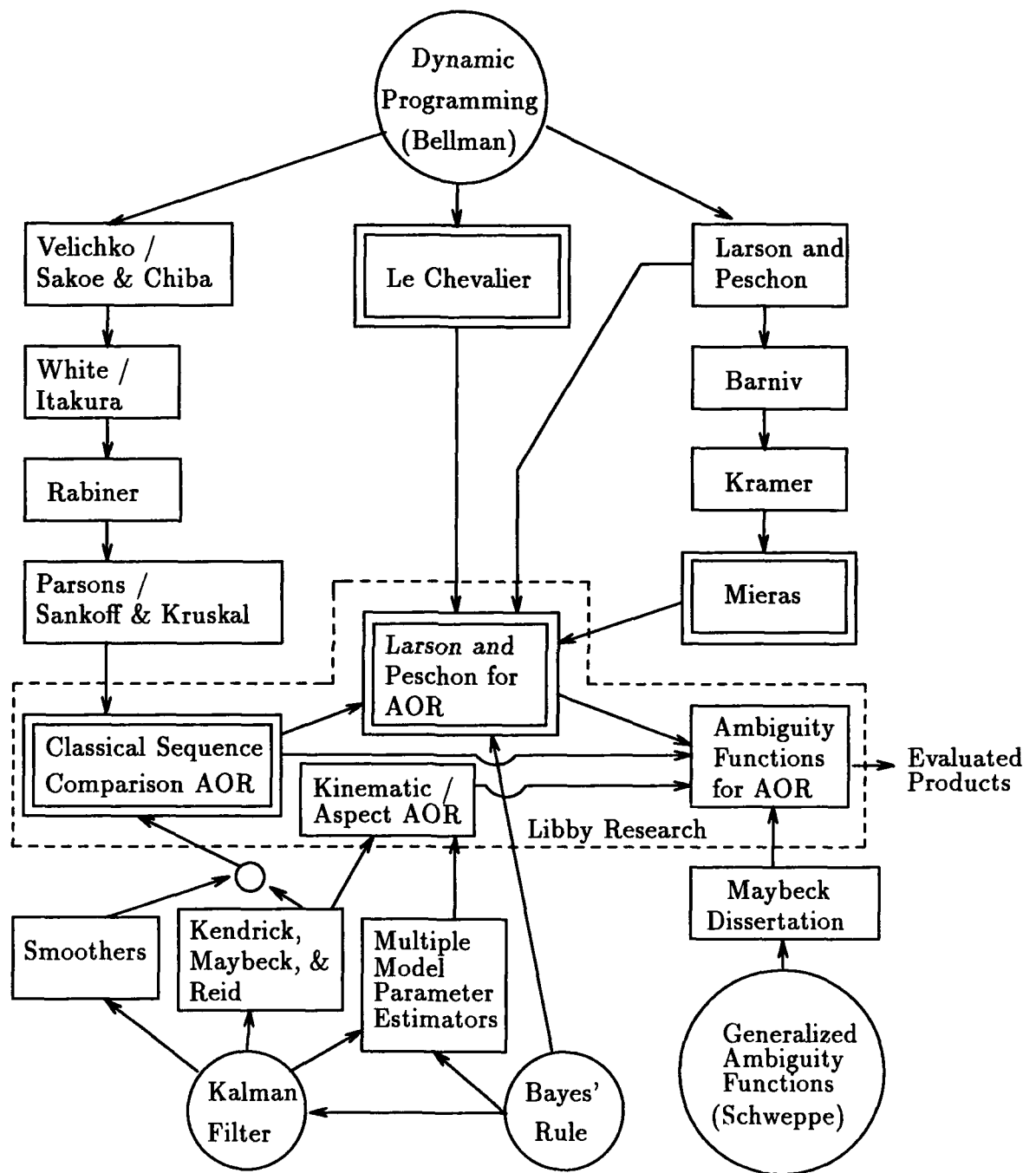


Figure 3.10. Relating Earlier Developments to the Author's Research (AOR: Automatic Object Recognition)

ing path (the author's original inspiration), lists researchers whose efforts were discussed in Sect. 2.4.2. The center path is Le Chevalier (Sect. 2.5.7), whose syntactic approach is also fundamentally language-inspired. The rightmost path is that begun by Larson and Peschon, whose equations were applied for tracking by Barniv, as described in a recent text on multisensor fusion [8], and which served through earlier sources [12, 13, 129] as the inspiration for Mieras (Sect. 2.5.8).

The lower portion of Fig. 3.10 refers to the origins of the linear or linearization-based estimation and estimator evaluation techniques applied in this research, as discussed in Sect. 2.3.1.1 and 2.7, respectively. It should be clear also that Bayes' Rule plays a key role in the basic Larson and Peschon equations, but this connection is not shown for reasons of clarity. The fundamental contribution of this research is the fusion of information from the upper and lower portions of this figure, using Bayesian methods where possible, to provide new understanding of, and new approaches for, dynamic object recognition.

*3.8.1 Contrasting Prior Efforts with the Author's Research.* We now address the relationship between the Larson and Peschon approach advanced in Sect. 3.6, on one hand, and the Le Chevalier and Mieras concepts on the other hand. Simply, it is clear that *both the Le Chevalier and Mieras concepts are sub-optimal implementations of the Larson and Peschon equations, in aspect angle space.*

Le Chevalier's development effectively reduces the Larson and Peschon conditional transition likelihood  $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$  to "evolutionary constraints" of unspecified form, and his measurement-to-possible origin state comparisons do not preserve the general probabilistic meaning of the Larson and Peschon term  $p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})$ , although Le Chevalier's inter-signal Chi-square metric is related to this form of a likelihood. Similarly, Mieras has reduced the transition likelihood to a hard "yes/no" association limit based on whether or not an earlier measurement/aspect angle pair (or terminus of a path of pairs) lies within the "association gate" of a later measurement/aspect angle pair (accounting evidently also for an aspect angle bias in the apparent direction of motion [162, 163]). Like Le Chevalier, Mieras uses a probabilistic measurement-to-state metric (as discussed in Sect. 2.2.3).

In effect, Le Chevalier and Mieras effectively replace the aspect angle state transition likelihood  $p(\mathbf{x}_{j+1,n} \mid \mathbf{x}_{j,n}, \mathbf{Z}_m^d, \omega_i)$  with a uniform probability density, the extent of which is defined by aspect angle transition bounds allowable by vehicle kinematics. For Le Chevalier, where the aspect angle transition bounds are evidently circular around each candidate current or prior aspect angle state, the uniform probability density can be further described as zero mean. For Mieras, the mean is biased in the apparent direction of motion. For both Le Chevalier and Mieras, the probability of starting cell location  $p(\mathbf{x}_{0,n} \mid \mathbf{Z}_m^d, \omega_i)$  is treated as equal for all starting states, or effectively ignored. This “fixed bound” approach is illustrated in Fig. 3.11, using the multiple-window format common to earlier figures in this chapter.

Recalling our comments in Sect. 2.6.2, we can see that this fixed bound approach is a reasonable restriction of the domain of the likelihood functions which match feature observable measurements to object signature libraries. Thus, fixed bound restrictions promise improved recognition performance over algorithms which do not restrict that domain. However, these fixed bound approaches cannot make full use of the *information content* in the transition likelihood  $p(\mathbf{x}_{j+1,n} \mid \mathbf{x}_{j,n}, \mathbf{Z}_m^d, \omega_i)$ . As we will see in Chapter V, the practical effect of this suboptimality is that, where high quality transition likelihood information is available, fixed bound algorithms have a greater tendency than more optimal approaches to allow *unlikely* aspect angle transitions. This means that, in comparison to more optimal approaches, fixed bound approaches will have a greater tendency to misclassify a target, or to estimate the aspect angle sequence on the true target incorrectly.

*3.8.2 Contrasting Classical Sequence Comparison-Based vs. Larson and Peschon-Based “Motion Warping” Concepts.* At this point, we can express the difference between classical sequence comparison-based and the Larson and Peschon-based “motion warping” concepts (including those of Le Chevalier and Mieras) in terms of how two sequences (observed and library model-derived) are compared to each other, and what information is used.

This comparison revolves around two points: recall the discussion in Sect. 2.4.5 to the effect that (1) the Larson and Peschon equations are an elaborate forward dynamic

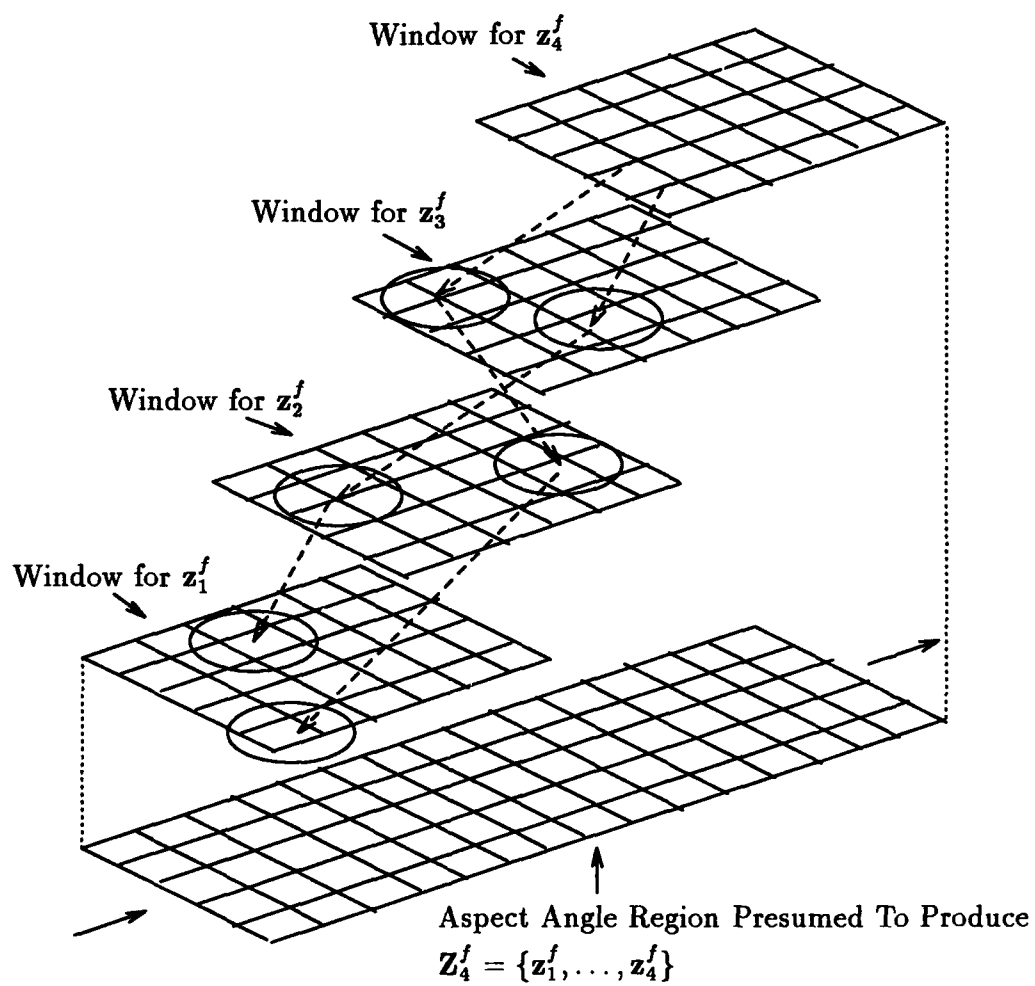


Figure 3.11. Typical Matching Path - Fixed Bound Algorithm



programming-based sequence comparison technique with a probabilistic metric and continuity rules, and (2) that the continuity rules in the Larson and Peschon equations are further restricted to require deletion of as many points as required from the library set to match the unknown set, point for point.

First, in classical sequence comparison-based motion warping, we “pre-warp” the aspect angle space using *a priori* kinematic information in a preliminary attempt to predict a sequence of observations, and then use dynamic time warping-like techniques to make the final comparison. In the Larson and Peschon-based concepts, we warp the observations to fit a like number of elements in the aspect angle space. Use of kinematic information in L&P techniques may vary from explicit inclusion of transition likelihoods at one extreme, to simple feasibility bounding at the other extreme.

More specifically, in classical sequence comparison the “pre-warping” process, i.e., laying out the aspect angle space along the kinematically-estimated nominal aspect angle path, has the effect of organizing the aspect angle space according to the maximum probable transition rate, or simply reordering the  $\mathbf{x}_k^a$  in time in accordance with the maximum  $p(\mathbf{x}_{k+1}^a | \mathbf{x}_k^a)$  for all  $\mathbf{x}_k^a$ . The motion warping process then considers deviations around this nominal. This is the key difference between classical sequence comparison and a Larson and Peschon-type (Le Chevalier/Mieras) approach, which works on an “unwarped” aspect angle space directly, considering  $p(\mathbf{x}_{k+1}^a | \mathbf{x}_k^a)$  or some less elaborate transition constraint *explicitly* in each transition decision.

Second, compare Figs. 3.3 and 3.4 to Fig. 3.8 to observe how classical sequence comparison in general forces every library signature along some contiguous aspect angle path to associate with the observed measurements, while L&P-based techniques can allow the algorithm to ignore unlikely associations. The potential pitfalls of this issue were observed in Sect. 3.6.6. The “basic” Larson and Peschon concept is a sequence comparison technique calling for  $m$  observations to be matched one-for-one with exactly  $m$  points on a model. Thus, as implied above, Larson and Peschon-type approaches (including those of Le Chevalier and Mieras) can result in picking a “nice” sequence of points from an otherwise inappropriate model. Classical sequence comparison, or path warping concepts, however,

may be less likely to make such an inappropriate choice, due to their tendency to allow association of *dissimilar* numbers of the elements of the two compared sequences [195].

These differences are perhaps indicative of a fundamental difference between the L&P approach and classical sequence comparison. Classical sequence comparison is based on comparing discrete observations of an underlying continuous or piecewise continuous feature space, at an observation frequency which (in a Shannon sampling theory sense) captures the spatial variational trends in the feature observable values. The L&P approach does not require that assumption, and indeed seems designed to make the best possible decision where that assumption cannot be made.

Finally, comparing classical sequence comparison side-by-side with a Larson and Peschon-type algorithm reveals a subtle possible practical advantage for the classical approach. Note that, in the one-dimensional classical algorithm with "basic" continuity constraints (Fig. 3.8), each aspect angle cell in each window has at most two possible predecessors, compared to Larson and Peschon (L&P)-type algorithms (including the Le Chevalier and Mieras algorithms), where predecessors for each aspect angle cell in each window may include many cells in the previous window (particularly some which imply "backward" motion of the object). Since dynamic programming computational load is driven largely by the number of states and predecessors (*dimensionality*), it is clear that classical "continuous" sequence comparison with simple continuity constraints may have a potential for reduced computational load.

*3.8.3 Comparing Use of Kinematic Information in the Author's Research with the Le Chevalier / Mieras Concepts.* Neither the Le Chevalier nor Mieras approaches make optimum use of the rich information available from kinematic measurements regarding aspect angle states and transitions, the approach that forms the core of the author's research.

Le Chevalier's writings do not specify how the "evolutionary constraint" parameters would be obtained for the target identification case. As noted in Sect. 2.5.7, his insistence on "real time" operation and his apparent unwillingness to model and propagate the target kinematic state imply that he does not use a smoother or target kinematic state estimator, and therefore has little knowledge of the likely aspect angle transition of the observed

target – only that that transition is bounded. In the Mieras approach, the “evolutionary constraints” are apparently fixed, but the constraints are evidently biased in the direction expected from kinematic tracking [162, 163].

In motion warping with the full Larson and Peschon approach, the “evolutionary constraints” are functions of the object kinematic state and state covariance estimates, as derived from an extended Kalman filter/smoothen operating as discussed in Sect. 3.6.5. This procedure will be described in full detail in Chapter V. Thus, kinematic information is maintained and fused in the same Bayesian probabilistic framework as feature observable information. With “classical” sequence comparison approaches, we at least constrain the matching process to follow aspect angle trajectories with maximum *a priori* likelihood  $p(\mathbf{x}_{k+1}^a | \mathbf{x}_k^a)$ .

**3.8.4 Advantages of Predecessor Approaches.** The reader must not be left with the impression that the Le Chevalier and Mieras approaches would be less effective in every case than full or “optimal” implementation of the Larson and Peschon approach or classical sequence comparison methodology. As with every class of pattern recognition algorithms, the decision to employ more or less complex implementations requires trade-offs considering feature space, information and time available, computational burden, and so on.

As our results will show, basic L&P approaches like those of Le Chevalier and Mieras can make a substantial discrimination improvement by restricting the wild aspect angle transitions (and unreasonable low-cost matches) attempted on incorrect object classes by an “independent-look” recognizer (i.e., a conventional matching algorithm, as described in Sect. 2.2.1) working in a noisy signature domain. However, these “aspect angle bound” algorithms can allow apparent aspect angle transitions that are inconsistent with the observed kinematics, such as aspect angle sequences that stop or move in the opposite direction from that implied by the observed kinematics – even when a set of observations is matched to the *correct* object class.

These effects are often exhibited in our tests, particularly for incorrect matches, and suggest that subsequent processing of these aspect angle sequences and/or inclusion of

additional (ideally independent) feature observable measurements in each  $z^f$  is warranted to improve classification. The Mieras algorithm is believed to apply the former, or sequence processing approach [162], which may be a significant improvement over the approach of Le Chevalier *et al.* [136]. It is important to note that the tests conducted by Le Chevalier *et al.* were conducted in a one-dimensional (great circle) aspect angle space, where this "wandering" would have been less noticeable.

The apparent advantage accrued by these "aspect angle bound" approaches is that, theoretically, they can be made to be "real time", since they do not require the time delay required to develop the kinematic measurement-derived aspect angle rate estimate, or  $p(\mathbf{x}_{j+1,n} \mid \mathbf{x}_{j,n}, \mathbf{Z}_m^d, \omega_i)$  given by a kinematic tracker/smoothing combination. As we noted above, however, for aircraft targets, body angular rates can be on the order of hundreds of degrees per second, and are unobservable to the kinematic trackers generally used to find "global" aspect limits or "windows" for feature observable-matching recognition algorithms. Thus, it seems clear that one may need to accept delays of up to a few seconds and some form of smoothing to provide any reliable aspect angle estimates from kinematic (translational) measurements alone. If the target is determined to be turning during this period, that kinematic information can and should be used explicitly.

Chapter V will contrast the performance of various conventional and dynamic programming-based pattern recognition approaches using generalized ambiguity functions. The following section describes how multiple model Kalman filter parameter estimators and dynamic programming sequence comparison methods can be combined to address shortcomings in each respective approach. The result is a new class of estimators that can fuse kinematic *and* ambiguous feature observable information *in real time*.

### 3.9 A New Class of "Coupled" Estimators for Object Recognition

In Sect. 3.4, we used Kendrick-type kinematic/aspect-angle estimators in a classical multiple model / residual analysis structure to give an expression for the probability of object class membership via Bayesian techniques, where the new information is given by kinematic measurements and aspect angle pseudo-measurements, conditioned (implicitly) on feature space measurements. Conversely, in Sect. 3.6, we used the Larson and

Peschon (L&P) approach and Bayesian methods to find an expression for the probability of object class membership, where the new information is given by feature space measurements, conditioned on the likelihood of aspect angle transitions, as implied by kinematic measurements.

We observe that Kendrick-type kinematic/aspect-angle estimators work poorly where inadequate estimates of aspect angle are provided, and conversely that L&P estimators work poorly where inadequate *a priori* information  $p(\mathbf{x}_{k+1}^a | \mathbf{x}_k^a)$  is provided. If these two forms of information processor are used *jointly*, however, we can use Bayesian techniques to obtain a *practical* expression for *the probability of class membership conditioned jointly on feature space and kinematic measurements*. This is a major objective of this dissertation research, and provides Step Three in the three step process outlined at the start of this chapter.

These last observations allow us to envision a new class of "coupled" filter structures, in which conventional filters and /or smoothers perform linear or linearized estimation tasks appropriate to them, and counterpart Larson and Peschon estimators perform nonlinear estimation tasks to which they are suited. The key point to recognize is that the L&P aspect angle path  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ , in fact provides a maximum likelihood aspect angle estimate for each object model  $\omega_i$  at any time  $t_k$ . This aspect angle information from the L&P estimator can be passed to the conventional filter as a pseudo-measurement, and information from the conventional filter is made available to the L&P estimator as "*a priori*" information. Judgments as to the "joint likelihood" of measurements and states can be made then, using information from both sources concurrently.

At any particular measurement time  $t_k$ , an estimate (*estimate* because we consider the L&P path only, as in Eqn. (3.18)) for the joint likelihood of measured kinematic and feature observables and selected state values is given by:

$$\hat{p}(\hat{\mathbf{X}}_{k/\omega_i}^{LP}, \mathbf{Z}_k^f, \mathbf{Z}_k^d, \hat{\mathbf{X}}_k^r | \mathbf{Z}_{k+s-1}^d, \mathbf{Z}_{k-1}^f, \omega_i) = \hat{p}(\hat{\mathbf{X}}_{k/\omega_i}^{LP}, \mathbf{Z}_k^f | \mathbf{Z}_{k+s-1}^d, \mathbf{Z}_{k-1}^f, \omega_i) \left\{ \prod_{n=1}^k [p(\mathbf{z}_n^d, \hat{\mathbf{x}}_n^r | \mathbf{Z}_{n+s-1}^d, \mathbf{Z}_{n-1}^f, \omega_i)] \right\} \quad (3.20)$$

so that, using Bayes' Rule, we obtain the desired *a posteriori* probability of class membership as:

$$\hat{p}(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_{k+s}^d) = \frac{\left\{ p(\hat{\mathbf{X}}_{k/\omega_i}^{LP}, \mathbf{Z}_k^f | \mathbf{Z}_{k+s-1}^d, \mathbf{Z}_{k-1}^f, \omega_i) \prod_{n=1}^k [p(\mathbf{z}_n^d, \hat{\mathbf{x}}_n^r | \mathbf{Z}_{n+s-1}^d, \mathbf{Z}_{n-1}^f, \omega_i)] \right\} p(\omega_i)}{\sum_{\omega_j, j=1}^J \left\{ p(\hat{\mathbf{X}}_{k/\omega_j}^{LP}, \mathbf{Z}_k^f | \mathbf{Z}_{k+s-1}^d, \mathbf{Z}_{k-1}^f, \omega_j) \prod_{n=1}^k [p(\mathbf{z}_n^d, \hat{\mathbf{x}}_n^r | \mathbf{Z}_{n+s-1}^d, \mathbf{Z}_{n-1}^f, \omega_j)] \right\} p(\omega_j)} \quad (3.21)$$

where all quantities are as defined earlier. Note, however, that since the Larson and Peschon transition likelihoods are now functions of previous feature observable measurements, the Larson and Peschon joint probability density expression  $\hat{p}(\hat{\mathbf{X}}_{k/\omega_i}^{LP}, \mathbf{Z}_k^f | \mathbf{Z}_{k+s-1}^d, \mathbf{Z}_{k-1}^f, \omega_i)$  now contains the term  $\mathbf{Z}_{k-1}^f$  as a conditioning argument, which it did not in Sect. 3.6.

It is important to note that the representation given in Eqn. (3.21) is but one in a class of such representations. First, following the comments in Sect. 3.6, we may choose to add contributions to the likelihood of the observed feature measurements from paths other than the L&P path  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ . Second, the choice of kinematic/aspect estimator models is completely open to the designer. The estimator to be demonstrated in Chapter IV is, for example, similar to but also significantly different from the Kendrick/Maybeck/Reid and Andrisani *et al.* estimators.

A key point about this recognizer is that theoretically, it can work in real time, unlike the recognizers discussed in Sect. 3.6.5 and Chapter V, which nominally must employ smoothers against highly dynamic objects (i.e., aircraft). To attain the same state estimation accuracy, however, such real-time designs would require in general considerably more complicated filter models than smoother-based designs.

### 3.10 Assessing the Proposed Recognizers as Syntactic Approaches

Recalling the discussion in Sect. 2.2.2, it becomes clear that each of the three proposed object recognition approaches is *syntactic* in nature. For the second and third approaches, employing classical or Larson and Peschon "variant" dynamic programming-based sequence

comparison, this relationship is made clear by the association of DP sequence comparison and syntactic approaches in Miclet [161] and Le Chevalier [136].

The first approach, using conventional filter residuals and related quantities only, is recognized to be syntactic when we consider that presentation of observed quantities *in the proper time order* to the  $J$  filters is absolutely essential to their function. As with an automaton classically used in syntactic recognition, the correct conventional filter model reveals itself by remaining in an acceptable "state" as it processes the observed sequence – this acceptable state is simply obedience to proper behavior of the residual sequence as discussed in Sect. 2.3.1.3. This observation is equivalent to that made by Therrien [211], in which he associated residual sequence analysis methods with classical syntactic theory [88].

### 3.11 Evaluating Algorithm Performance

**3.11.1 Introduction.** This section provides approaches for evaluating the performance of the object recognition algorithms proposed in this chapter. In particular, we wish to motivate the use of the generalized ambiguity function (GAF) for evaluating algorithm performance, and for providing insights not available from conventional evaluation techniques.

**3.11.2 Conventional Performance Evaluation.** The conventional approach to evaluating object or target recognition algorithms is to (1) define objects or targets of interest, (2) obtain real or simulated sensor data for these objects, (3) choose a set of noise-corrupted measurements from one particular object to represent a true, unclassified *a priori* object, and (4) evaluate the ability of any given algorithm to identify correctly the object that generated those measurements. A correct identification event may be defined as one in which the value output by an algorithm "tuned" for the correct object (unknown *a priori*) is higher than any of the values output by the same algorithm "tuned" for incorrect object classes. Conversely, an incorrect identification is one in which an improperly-tuned algorithm gives a higher output value than that of the properly-tuned algorithm. Thresholds may be established for the differences between two or more recognizers to meet some confidence level prior to making a decision.

This approach typically quantifies performance in terms of percentage of correct or incorrect identification for any algorithm, true object, measurement set, and false object set. The "best" algorithm is typically taken to be the one which has the highest percentage or probability of correct identification for some particular set of objects and scenarios. Each of the approaches considered earlier in this chapter can be evaluated in this fashion.

*3.11.3 Performance Evaluation with the Generalized Ambiguity Function.* On the other hand, the generalized ambiguity function, as discussed in Sect 2.7, presents an entirely different approach for analyzing object recognition system performance. To understand this, we must relate the parameter estimation concepts of (1) states, (2) parameters, and (3) measurements to the corresponding forms appropriate for object recognition. Then, we will consider object recognition algorithms as generators of likelihood functions, the mean values of which for particular measurement sets and scenarios define generalized ambiguity functions. Finally, we will discuss how generalized ambiguity functions demonstrate improved recognition through fusion of kinematic and sensor signature information.

All object recognition algorithms can in fact be interpreted as likelihood functions in the sense used by Rao [184:353] and Maybeck [154:75]—whether or not their outputs provide the *classical* likelihood value  $p(\mathbf{z} | \Omega)$ , i.e., the probability density of some measurement  $\mathbf{z}$  given that the state and parameter set  $\Omega$  is being observed. As discussed in Sect. 2.6.2, all that we require is that the output value of a likelihood function  $L$ , defined for a particular set  $\Omega$  of states and parameters, operating on measurements from a process with *that* set of states and parameters, should be greater than the output value of any analogous likelihood function defined for *another* set of states and parameters, operating on the (same) measurement set from process  $\Omega$ .

The particular form of object with which we are most concerned in this research is the fixed wing aircraft. The concepts of "states" in classical aircraft recognition and parameter estimation do not differ widely. In aircraft recognition, the fundamental states of interest are the six degrees of freedom — translation and rotation with respect to some reference (and higher derivatives of these quantities) — for the target aircraft. Taken together with the corresponding states for the sensor platform, these states dictate the appearance of the



target aircraft to the sensor. Other "states" may be of interest as well – engine speed and temperature, orientation of objects on the target relative to the target body frame, and so on. Similarly, the concepts of measurements in aircraft target recognition and parameter estimation are identical.

The concept of "parameters" in object recognition, however, is not well defined – extending this concept is key to this research. Intuitively, a real object must only be representable as an infinite dimensional vector, and there may be any number of ways to define "basis vectors" or individual dimensions in the corresponding vector space. For example, for aircraft targets, one might define an abstract parameter space based on classical aircraft design features – fuselage length, diameter, wingspan, and so on. Clearly, such a parameter space could have both continuous and discrete (e.g., number of engines) attributes. One set of basis vectors might serve as well as some other to define objects and behavior of interest. A particular question arises immediately regarding fidelity of necessarily finite-dimensional models versus infinite-dimensional truth – how many parameters of what kind are required for a model to achieve a given level of closeness (with respect to some metric) to true behavior? This research does not address that question *per se*, but will motivate the need for further research in that direction in Chapter VI.

Classical object recognition evaluation approaches, as discussed in the previous section, sidestep the question of parameter spaces by considering only *discrete points* in that space corresponding in some sense to *known* objects. Classical approaches evaluate the performance of recognition algorithms at these points, but not elsewhere.

In evaluating a recognition algorithm, however, the performance of that algorithm against objects in some sense "in between" real objects of interest should be important as well. For example, in general we should probably prefer (1) a "robust" algorithm that returns high likelihood function values for objects *close* in some sense to the design object, over (2) an algorithm which fails utterly (returns low likelihood values) when presented with measurements from a object with only minor variations from the design object. On the other hand, if we truly wish to identify small variations from some design point, such a robust algorithm may be utterly inadequate. Consider the case in which we desire to distinguish an F-4G ("Wild Weasel" air defense suppression variant) Phantom II from

an F-4E (standard multirole tactical fighter) Phantom II: a robust algorithm would be counterproductive.

The intention here is not to propose definitive models or rules for considering object parameter spaces. The point is simply that the use of the generalized ambiguity function in object recognition *forces* one to consider the existence and significance of these spaces, and provides a natural tool for evaluating recognition algorithm performance as the algorithms – likelihood functions – are “tuned” over various domains in these spaces. In Chapter VI, we will consider other implications of the concept of object parameter spaces.

The aircraft model format used in Chapter V to evaluate “motion warping” algorithms offers one natural approach for defining “pseudo-objects” or object parameter sets in some sense *in between* real objects of interest. That chapter will discuss the definition of particular objects and “pseudo-objects” of interest. Likelihood function values are then defined for each of these points, and output values are found for each likelihood function operating on measurement sets from a “true” object. Note that likelihood functions are distinguished here from one another according to (1) their form or structure (the way they use information); and for functions with the same structure, (2) the particular object signatures they are tuned to identify.

It is important to note that this will entail a “Monte Carlo” evaluation of the generalized ambiguity function – analytical evaluations of the integral expression Eqn. (2.37) for a “motion warping” likelihood function appear intractable, due to the presence of numerous nonlinearities. For scenarios in which a completely linear, Gaussian description of the joint conditional density  $f_{Z|\Omega_t}(Z | \Omega_t)$  could be obtained, however, an analytical evaluation of the ambiguity function (Eqn. (2.37)) for this classical likelihood function (in natural log form) also could be obtained, as in [152].

Now, consider contrasting (1) the ambiguity function for an object recognition algorithm which fuses kinematic and feature observable information with (2) the ambiguity function for an object recognition algorithm which uses feature observable information only. How should they differ?

The discussion in Sect: 2.6.2 answers this question – when a set of measurements is matched to the wrong object model, or “point” in parameter space, a kinematic/feature observable fusion algorithm is more likely to yield *lower* likelihood function values than one which does not consider kinematics. With reference to Eqn. (2.37), we consider that as the parameters  $\mathbf{y}$  in  $\Omega$  change, a likelihood function *restricted* by kinematics will have a smaller chance of (improperly) finding a high likelihood function value somewhere in a large allowable aspect angle extent than a likelihood function which is not restricted by kinematics.

Since the generalized ambiguity function is the mean value of the likelihood function, then, the kinematic/feature observable fusion algorithm should provide a more sharply peaked, or less “broad” generalized ambiguity function, having greater curvature at the correct parameter value than methods that do not fuse information from feature observables *and* motion. This reduced ambiguity implies increased discrimination capability for the recognition algorithm which fuses kinematic and feature observable information. Chapter V will show this behavior graphically.

Finally, recall from Sect. 2.7 that the curvature of the generalized ambiguity function at the correct parameter value is directly related to the Cramér-Rao lower bound for the covariance of a parameter estimate by that likelihood function. This means that by choosing to consider continuous parameter spaces in object and target recognition, and by using the generalized ambiguity function, we can define the Cramér-Rao lower bound as a measure of relative performance for our recognition algorithms – whether or not the likelihood function is the classical likelihood function  $p(\mathbf{z} \mid \omega_i)$ . In classical target recognition at least, this approach has evidently never been proposed.

### 3.12 Chapter Summary

The goals of this chapter were to propose new approaches for recognition of dynamic objects in general and aircraft targets in particular. These new approaches were advanced by defining estimators to provide: (1) the likelihood of kinematic measurements and pseudo-measurements conditioned on feature observables, (2) the likelihood of feature observable measurements conditioned on kinematic measurements, and (3) the *joint*

likelihood of kinematic *and* feature observable measurements. We then used these likelihoods with Bayes' Rule and *a priori* probabilities to propose object recognition algorithms. Finally, we considered tools for evaluating the performance of these algorithms.

The following two chapters will show results from algorithms of the first two types, indicating the potential for improved object recognition from all three approaches, by comparison where possible with classical techniques that do not fuse kinematic and feature observable information. Both classical techniques and generalized ambiguity functions will be used to assess performance.

The theoretical and practical contributions of the effort described in this chapter are major. They encompass the essence of this dissertation, and include:

- (1) Extension of conventional multiple model residual sequence analysis techniques and kinematic/aspect-angle trackers to provide new methods for object and target recognition, in particular where sensor measurements are not linearly predictable.
- (2) Extension of the Larson and Peschon equations to provide new methods for object recognition using measurements from ambiguous feature observable spaces, considering *a priori* information from kinematics and other sources as to the likelihood of transitions on the underlying aspect angle state space.
- (3) Extensions of the theory and practice of classical sequence comparison to include feature observable sequences arising from an aspect angle subspace.
- (4) Combination of the Larson and Peschon equations with conventional linear estimators to provide a new form of estimator, suitable in particular for object recognition with ambiguous feature observables, generated from dynamic subspaces that exhibit linear behavior in some respects.
- (5) Through contributions (1) through (4) and application of Bayes' Rule, several new approaches for multisensor fusion to obtain an *a posteriori* probability of object class membership, conditioned jointly on kinematic and "nonkinematic" or feature observable information and *a priori* information for each known object class.
- (6) Identification of a new method for evaluating object recognition algorithms – the generalized ambiguity function.

(7) Extension of classical parameter space concepts into the field of object and target recognition.

(8) Through contributions (6) and (7), identification of a practical approach for obtaining a Cramér-Rao lower bound for dynamic object and target recognition algorithms.

The remainder of this dissertation will demonstrate and elaborate upon these developments. Chapter VI proposes extensions to the author's efforts.

## IV. Residual Analysis for Sensor Fusion and Object Recognition

### 4.1 Introduction

In this chapter, we will propose a kinematic/aspect-angle tracking filter design for aircraft target recognition, and use it for residual/state analysis and Bayesian parameter estimation methods, as discussed in Sect. 2.3.1.3. The purpose of this effort is to demonstrate "Step One" as proposed in the introduction to Chapter III – the use of multiple model/residual sequence analysis approaches for recognition of objects having "coupled" state dynamics models, using kinematic and feature observable measurements, without *explicitly* considering residuals in the feature observable space.

The efforts in this chapter were inspired originally by the simple observation that the performance of Kendrick *et al.* and Andrisani *et al.* trackers (see Sect. 2.3.3.1) must be *very* sensitive to proper choice of target model. Using the classical residual sequence analysis approach, we will exploit that sensitivity by noting that when a particular association of (1) measurements from an unclassified target and (2) filter target model *fails* to provide expected residual and/or state behavior, that association is suspect. Conversely, the association that exhibits the "best" residual/state behavior may be taken to indicate the correct target class. Finally, information of this kind from many different measurement sources and particular states can be treated using Bayes' Rule for proper probabilistic "weighting" to obtain a maximum *a posteriori* estimate of target class.

The proposed filter design is by no means uniquely suited to this application – the distinguishing attribute of the filter shown here is extreme simplicity. It is designed more to *fail* well when it should fail than to *track* well in a variety of situations – the key point is that *robustness* to incorrect target model choices is *not* a virtue in this application.

### 4.2 A Tracking Filter for Target Recognition

**4.2.1 Design Philosophy.** In App. C, the reader will observe that the Kendrick *et al.* and Andrisani *et al.* kinematic/aspect filter designs require a reasonable amount of target class-specific information that can only be obtained by flight control analysis or

empirical identification based on observing the respective target classes in flight. For this effort, this level of information was neither available nor required.

Consistent with the approach in the previous chapters, we will restrict this filter to operate under the assumption that the target flies in a straight (not necessarily level) path or in a coordinated turn with constant acceleration relative to the target body frame. For the short periods of time (several seconds) over which we intend to use this tracker, these are considered to be valid assumptions – in any case, we will discuss actions to be taken in the event that these assumptions appear to fail. The target's key turn-defining states will be modelled as constants ("driven", or influenced to change during propagation, only by small amounts of "continuous" white noise in the usual manner described in Sect. 2.3), and assumed to be unknown at the start of the tracking sequence – the filter's first task after initiation is to converge to reasonable estimates for these states. These key turn-defining states are the roll angle and angle of attack for the turn, relating in the classical nonlinear fashion to target velocity and position relative to the inertial frame (as discussed in Sect. 5.5.3).

The "confusion" experienced by this filter when true and filter target models are mismatched will be due primarily to significantly incorrect target rotation state estimates, driven by aspect angle pseudo-measurements given to the filter. Recall that the term *pseudo* applies because these quantities come from a pose estimator, rather than directly from a sensor *per se*. As in the Kendrick and Andrisani efforts, the proposed filter will place a great deal of reliance on these aspect angle pseudo-measurements in defining the target acceleration state estimate. Ultimately, for the wrong target model choice, the mismatch between (1) estimated acceleration (driven by the incorrect pose estimate) on one hand, and (2) estimated velocity and position (driven by reasonably good doppler velocity and position measurements) on the other hand, will cause significant residuals and/or unreasonable values for some states – betraying this incorrect model choice in the classical fashion discussed in Sect. 2.3.1.3.

We will now discuss how these mismatches can arise in practice, and how they affect filter processes. Note Fig. 4.1 during this discussion. First of all, assume that we are tracking an unrecognized aircraft target with radar and some (notional) imaging sensor

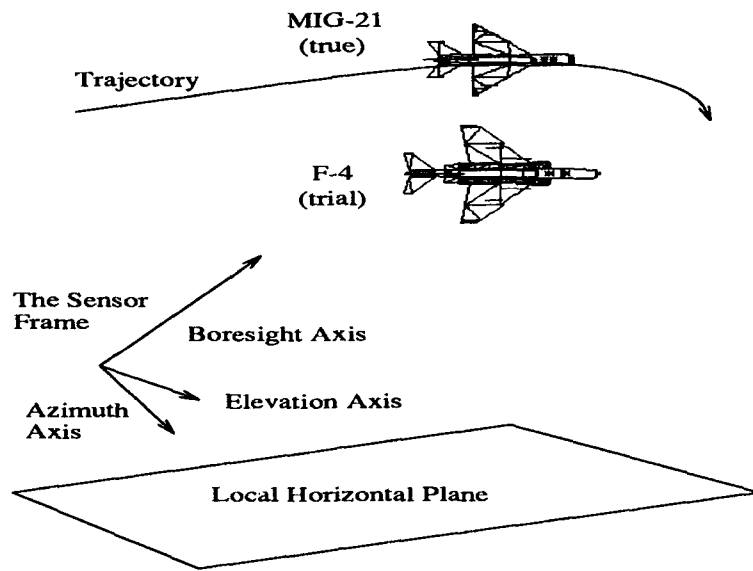


Figure 4.1. Tracking / Recognition Scenario for Discussion

as it turns from a crossing trajectory toward the sensor platform, as if to launch a missile at the platform. Simply tracking the target as a point object using the radar gives a good estimate of target position and velocity states – the imaging sensor will provide a pose estimate (and could improve cross-range position and velocity estimates if desired, although this is not essential to the discussion).

Finally, assume that the true target is a MIG-21, but that one of our target recognition alternatives requires us to try associating the target image with that of an F-4. This choice of candidate targets is *not* accidental – in image feature observable domains alone, even conditioned to some extent on kinematics (e.g., limiting aspect angle search windows according to observed motion), these two aircraft classes can be confused easily [228:77]. We will see, however, that in the kinematic state domain, conditioned on image information, they are readily distinguishable under common conditions.

Note that the F-4 is quite a bit longer than the MIG. Since range to the target is reasonably well known, our notional recognizer will presumably make use of that length difference (i.e., with reference to App. B.3, the recognizer is not, and should not be, *scale-invariant*). In order to fit a model or statistical library representation of an F-4 to the



image of an actual MIG, the pose estimator will in effect “pitch up” (or down) the F-4 model, decreasing its apparent length to match the image of the unclassified target (MIG). The problem now is that the resulting pose estimate, delivered to the kinematic/aspect tracker, will imply an extremely large angle of attack for the “F-4”. This further implies a much higher acceleration *magnitude* than the target is actually executing. Over time, this unreasonable acceleration estimate will cause inconsistency within the filter, as position and velocity estimates develop based on kinematic measurements.

Moreover, the wingspan of the F-4 is quite a bit larger than that of the MIG. To resolve this difference, the pose estimator will presumably in effect “roll” the F-4 model, decreasing its apparent width to match the image of the unclassified target. This error in the pose estimate will imply that the plane of the wings lies at a considerably different orientation than does the true orientation, which will lead to large errors in the estimated *direction* of the target acceleration. This factor too will lead to inconsistency within the filter, as we shall see.

Generically, the pose estimate errors described thus far are “bias” errors. Other sensors – high range resolution (HRR) radar and so on – should also be expected to provide pose estimates exhibiting biases due to incorrect model choices. For example, the Mahalanobis HRR metric used in Chapter V is extremely sensitive to length (in range) of the HRR signature – as in the previous example, a pose estimator using this metric can be expected to “turn” a *long* library model to find a best fit for the (*short*) signature from a smaller actual target (note that the words “long” and “short” are italicized because HRR signature “length” is a function of multi-bounce and other effects, as well as target physical size, as discussed in Sect. 2.2.3).

Other forms of errors may be expected from any pose estimator, like erratic variations between measurements (more akin to “white” measurement noise errors), or time-correlated errors (e.g., relatively slow wandering in aspect angle, which can be modelled as the outputs of integrators driven by white Gaussian noise). These “classical” error types are described in [153:183]. For radar-derived pose estimates in particular, we might expect “white”-type errors due to scatterer motion that would affect pose estimates for correct or incorrect model choices.

Bias errors due to incorrect target model choices and white noise from various sources are considered to be well-representative of typical errors expected in actual target recognition scenarios. Moreover, from a theoretical perspective, bias and white errors tend to "bound" the different forms of error in general. For these reasons, these classes of errors will be the only ones considered in this demonstration.

**4.2.2 The Filter State Model.** The state dynamics model for the proposed extended Kalman filter is given in the equation:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p}_{t/i_N} \\ \dot{p}_{t/i_E} \\ \dot{p}_{t/i_D} \\ \dot{v}_{t/i_N} \\ \dot{v}_{t/i_E} \\ \dot{v}_{t/i_D} \\ \dot{\alpha}_b \\ \dot{\alpha}_c \\ \dot{\alpha}_{gm} \\ \dot{\rho}_c \\ \dot{\rho}_{gm} \\ \dot{\beta}_{gm} \\ \dot{k}_{aero} \end{bmatrix} = \begin{bmatrix} v_{t/i_N} \\ v_{t/i_E} \\ v_{t/i_D} \\ -k_{aero} f_{if} \mathbf{C}_{if:1,3}^i \\ -k_{aero} f_{if} \mathbf{C}_{if:2,3}^i \\ (-k_{aero} f_{if} \mathbf{C}_{if:3,3}^i) + g \\ 0.0 \\ 0.0 \\ -\frac{1}{\tau_\alpha} \alpha_{gm} \\ 0.0 \\ -\frac{1}{\tau_\rho} \rho_{gm} \\ -\frac{1}{\tau_\beta} \beta_{gm} \\ 0.0 \end{bmatrix} + \mathbf{G} \begin{bmatrix} w_{v_N} \\ w_{v_E} \\ w_{v_D} \\ w_{\alpha_b} \\ w_{\alpha_c} \\ w_{\alpha_{gm}} \\ w_{\rho_c} \\ w_{\rho_{gm}} \\ w_{\beta_{gm}} \\ w_{k_{aero}} \end{bmatrix} \quad (4.1)$$

where:

$p_{t/i_{N,E,D}}$  = position of the target in inertial frame coordinates, i.e., with components taken along the North, East, or Down axes of an earth-surface inertial or *navigation* frame. Dot notation denotes time rate of change of the indicated variable, as observed from and coordinatized in the indicated frame.

$v_{t/i_{N,E,D}}$  = velocity of the target relative to the inertial frame in inertial frame coordinates, i.e., as observed from and coordinatized in the navigation (inertial) frame.

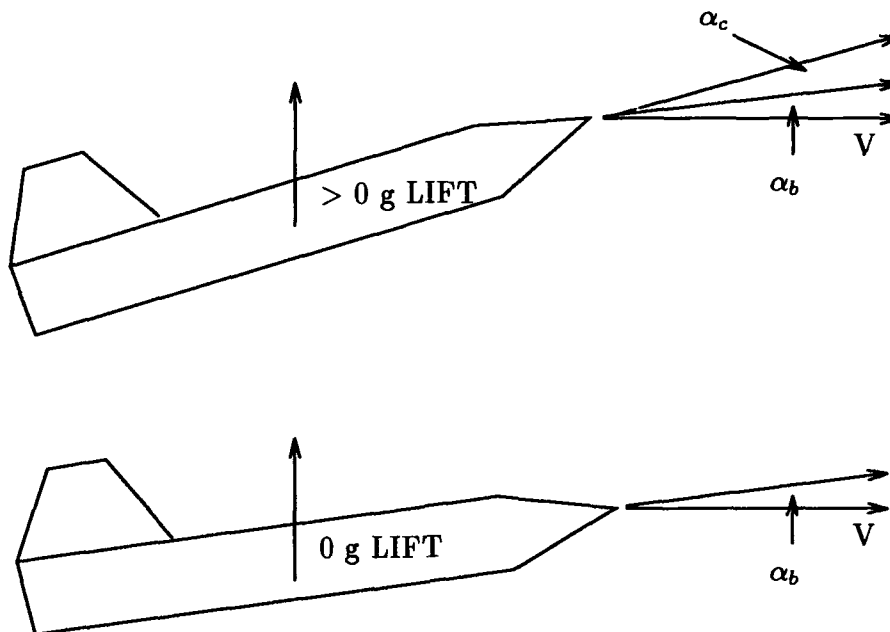


Figure 4.2. Distinction Between  $\alpha_b$  and  $\alpha_c$

$\alpha_b$  = the angle between the zero-lift longitudinal axis of the target and the target body longitudinal reference axis used by the target recognition algorithm. This angle is assumed to be constant for any given encounter, a function of the aircraft structure and trim conditions, and all of the factors that affect trim conditions (fuel status, stores, etc.). The degree to which this quantity is considered "known" or not by the filter can be varied by adjusting the state's initial filter covariance. Unlike the following two variables  $\alpha_c$  and  $\alpha_{gm}$ , which enter into lift force equations,  $\alpha_b$  does not, thereby mitigating what would otherwise be a serious observability problem, particularly between  $\alpha_b$  and  $\alpha_c$ . The difference between  $\alpha_b$  and  $\alpha_c$  is shown in Fig. 4.2.

$\alpha_c$  = the angle between the zero-lift longitudinal axis of the target and the angle of attack required to achieve the desired (pilot-commanded) lift. This angle is treated as a constant over the period of a few seconds required for this algorithm to function, reflecting the fact that aircraft turns are generally held at a near-constant turn rate for several seconds. As discussed below, this quantity is treated as unknown initially (i.e., this state has a high initial filter covariance), and can be "reset" to unknown (by again

artificially increasing the state's filter covariance) when filter residuals indicate a change in turn state for all models – this is an online adaptive estimation approach.

$\alpha_{gm}$  = the angular difference between the nominal angle of attack  $\alpha_c$  required to achieve the desired maneuver (turn) and the actual angle of attack, modelling small perturbations about the nominal that tend to null to zero quickly. (It is modelled as the output of a first-order Gauss-Markov process – hence the subscript “gm”.)

$\rho_c$  = the roll angle between the vertical “wings-level” attitude and the angle required to attain the desired orientation of the plane of the wings, as shown in Fig. 5.24. Treated as a constant for the same reason, and in the same way, as  $\alpha_c$ , noted above.

$\rho_{gm}$  = the angular difference between the roll angle required to achieve the desired turn direction and the actual roll angle, modelling small perturbations about the nominal that tend to null to zero quickly. (It is modelled as a the output of a first-order Gauss-Markov process – hence the subscript “gm”.)

$\beta_{gm}$  = the angular difference between the nominal zero sideslip angle and the actual sideslip angle, modelling small perturbations about the nominal that tend to null to zero quickly. (It is modelled as a the output of a first-order Gauss-Markov process – hence the subscript “gm”.)

$g$  = acceleration due to gravity

$f_{lf}$  = aerodynamic load factor normal to the velocity vector, or, equivalently, acceleration due to lift force, as computed in Eqn. (5.1). It is important to note that the  $\alpha$  or angle of attack argument used to compute  $f_{lf}$  is the sum of  $\alpha_c$  and  $\alpha_{gm}$ :  $\alpha_b$ , as noted in the definition of this variable above, accounts for reference differences or trim conditions only, and does not contribute to aerodynamic lift.

$k_{aero}$  = a multiplicative (scaling) factor for aerodynamic load factor (lift acceleration)  $f_{lf}$  which will be estimated to account for uncertainties in the factors of Eqn. (5.1) – including primarily aircraft mass, coefficient of lift, wing surface area, and, to a lesser extent, velocity and other factors. It is treated as a constant in the same way as  $\alpha_c$ , noted above.

$C_{ij}^i$  = the direction cosine matrix to convert a vector in "lift frame" coordinates to one in inertial frame coordinates. The lift frame is found by rotating the velocity frame (Fig. 23) through the roll angle  $(\rho_c + \rho_{gm})$  and defines the orientation of the normal load acceleration with respect to inertial space. The procedures for defining this direction cosine matrix (and others) are found in App. C. The two numbers associated with each term of this variable in Eqn. (4.1) are respectively the row and column required for the particular *scalar* element from  $C_{ij}^i$  used at that point.

$\tau_\alpha$  = correlation time for the first order Gauss-Markov process modelling the behavior of  $\alpha_{gm}$ . This quantity would be estimated theoretically or empirically for each target class.

$\tau_\rho$  = correlation time for the first order Gauss-Markov process modelling the behavior of  $\rho_{gm}$ . Estimated theoretically or empirically for each target class.

$\tau_\beta$  = correlation time for the first order Gauss-Markov process modelling the behavior of  $\beta_{gm}$ . Estimated theoretically or empirically for each target class.

$w_{v_N, v_D, v_D, \alpha_b, \alpha_c, \alpha_{gm}, \rho_c, \rho_{gm}, \beta_{gm}, k_{accro}} =$  appropriate continuous time (heuristically) zero-mean white Gaussian process driving noises, with appropriate strength  $Q(t)$  as defined in Sect. 2.3.1 and discussed in Sect. 4.2.4 below.

and :

$$G = \begin{bmatrix} 0_{3 \times 7} \\ I_{10 \times 7} \end{bmatrix} \quad (4.2)$$

These particular states were chosen because they reflect the coupling between translational and rotational dimensions for a conventional aircraft more simply and directly than many, if not all, other representations [120, 121, 5]. A more conventional representation for rotational states, such as a set of Euler angles relative to an inertial frame, contains the same information but requires more complicated transformations to relate angular state to translational accelerations. A shortcoming in the representation here is the treatment of angular perturbations as Gauss-Markov processes – additional aircraft-peculiar parameters as discussed in App. C would allow better modelling, but that level of detail was not required for this effort.

**4.2.3 The Filter Measurement Model.** The measurement model for the proposed extended Kalman filter is given in Eqns. (4.3) and (4.4) below. We assume that measurements of sensor-to-target range, (pointing) angle, and range rate are generated in the usual fashion by a radar tracker, and that range and angle are provided to the filter as target position in *sensor* frame coordinates (assumed instantaneously inertial). We assume that the pose estimator provides estimates of the three target body Euler angles relative to the filter inertial (*navigation*) frame, which are then processed relative to the target velocity vector *estimate* to yield pseudo-measurements of (1) angle of attack, (2) roll angle and (3) sideslip angle for input to the filter.

The *sensor* or *predicted line of sight* (pls) frame is a right-handed Cartesian reference frame defined by (1) the *predicted* sensor-to-target (boresight) vector, (2) the perpendicular (elevation) axis lying in a plane parallel to the local horizontal, and (3) the remaining (azimuth) axis, pointing generally down. This relationship is illustrated in Fig. 4.1. The orientation of the local horizontal (i.e., the direction of gravity) and magnitude of gravity are assumed to be perfectly known. The implications of this assumption of perfect gravity knowledge will be discussed at the end of this section.

The relationship between the (filter state or navigation frame) inertial and sensor frames is given by the direction cosine matrix  $C_i^{pls}$ . Procedures for defining  $C_i^{pls}$  are found in App. C. Note that, although this matrix is itself a function of the filter states, it is assumed to be *constant* at any measurement event (i.e., the sensor frame is impulsively updated – an appropriate assumption for this frame, which will be artificially maintained in software, even in actual implementation).

As discussed in Sect. 2.3.2.1 and [35], by comparison with other alternatives, this sensor frame approach offers several advantages. First, it is *inertial* between impulsive updates. Second, since range and angle error for our cases of interest are largely a function of target extent independently along each axis in the *true* (but unknown *a priori*) line-of-sight frame, and the predicted line-of-sight frame (pointing at the predicted target location) is the best estimate of that true frame, it is reasonable to treat the expected measurement error statistics in the predicted frame as independent along each axis (or diagonalized, in matrix form). Both the inertial nature and independent error statistics of this frame bring

computational savings in the tracker, with little effort or computational penalty required for impulsive updates.

In computing true and filter-estimated target locations, it is important to separate computations, in particular to prevent the filter from using truth knowledge, which an actual system could not access. In this simulation, position measurements treat the true target as a point, adding independently generated Gaussian noise samples to each axis in the *true* line-of-sight frame (for the reason noted in the previous paragraph). The point in space defined by this corrupted location in the true frame is transformed into the *predicted* line-of-sight frame, as it would be "seen" by the sensor. For simulation purposes, the true frame is generated in the same fashion as the *sensor* frame – their only difference is that the true LOS frame points at the true target location (unknown to the filter), whereas the sensor frame points at the best estimate for the target location. The reference frame drawing in Fig. 4.1 applies generally to either the *true* or *predicted* line-of-sight frames.

The measurement error (ellipsoid) statistics are also maintained separately for use as appropriate in the true or filter-modelled sensor frames. As for the position measurements, the range-rate measurement is generated relative to the true line-of-sight frame, but is assumed by the filter to lie along the boresight axis in the predicted line-of-sight frame.

Finally,  $z(t_i)$ , the measurement at time  $t_i$ , is modelled as the sum of  $h[\mathbf{x}(t_i)]$  (nonlinear form due to the consideration of ownship states not modelled in the filter state vector  $\mathbf{x}$ ) and a vector of discrete time zero mean white Gaussian noise  $\mathbf{v}(t_i)$ , with an appropriate covariance  $\mathbf{R}(t_i)$  (see Eqn. (2.15)), or:

$$\mathbf{z}(t_i) = \mathbf{h}[\mathbf{x}(t_i), t_i] + \mathbf{v}(t_i) \quad (4.3)$$

where:

$$\mathbf{h}[\mathbf{x}(t_i)] = \begin{bmatrix} p_{t/a_B} \\ p_{t/a_A} \\ p_{t/a_E} \\ v_{t/a_B} \\ \alpha_m \\ \rho_m \\ \beta_m \end{bmatrix} = \begin{bmatrix} c_i^{pls:b}(p_{t/i} - p_{a/i}) \\ c_i^{pls:a}(p_{t/i} - p_{a/i}) \\ c_i^{pls:e}(p_{t/i} - p_{a/i}) \\ c_i^{pls:b}(v_{t/i} - v_{a/i}) \\ \alpha_b + \alpha_c + \alpha_{gm} \\ \rho_c + \rho_{gm} \\ \beta_{gm} \end{bmatrix} \quad (4.4)$$

and:

$p_{t/a_B, A, E}$  = the position components of the target relative to the sensor (attacker) along the Boresight, Azimuth, and Elevation axes in the sensor frame, respectively

$v_{t/a_B}$  = the velocity component of the target relative to the sensor (attacker) along the boresight axis in the sensor frame, as provided by a doppler radar

$\alpha_m$  = the aspect angle sensor-derived pseudo-measurement of angle of attack, found by comparing the Euler angles of the target body frame relative to the filter inertial frame with the filter-estimated target velocity vector under the coordinated turn flight assumptions of Sect. 5.5.3

$\rho_m$  = the aspect angle sensor-derived pseudo-measurement of roll angle, found as for  $\alpha_m$

$\beta_m$  = the aspect angle sensor-derived pseudo-measurement of sideslip angle, found as for  $\alpha_m$

$c_i^{pls:b,a,e}$  = the first, second, and third rows (row vectors – hence use of lower case “c”) of the direction cosine matrix  $C_i^{pls}$ , which takes vectors from the filter inertial frame into the sensor (predicted line-of-sight) frame, consisting of boresight, azimuth, and elevation axes, as noted above

$p_{a/i}$  = position of the sensor (attacker) in the filter inertial frame, i.e., with components taken along the north, east, or down axes of an earth-surface inertial or navigation frame. Sensor position is assumed to be “perfectly” known (i.e., errors in target position are orders of magnitude greater than errors in ownship position)



$\mathbf{v}_{a/i}$  = velocity of the sensor (attacker) relative to and coordinatized in the filter inertial frame, i.e., with components taken along the north, east, or down axes of an earth-surface inertial or *navigation* frame. As for sensor position, sensor velocity is assumed to be "perfectly" known

and all other quantities are defined above.

The matrix of partial derivatives of  $\mathbf{h}[\mathbf{x}(t_i)]$  with respect to the states,  $\mathbf{H}$ , is therefore given by:

$$\mathbf{H} = \begin{bmatrix} & [\mathbf{C}_i^{pls}]_{3 \times 3} & & [\mathbf{0}]_{3 \times 10} & & \\ \mathbf{C}_{i:1,1}^{pls} & \mathbf{C}_{i:1,2}^{pls} & \mathbf{C}_{i:1,3}^{pls} & & [\mathbf{0}]_{1 \times 10} & \\ & & & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ & & [\mathbf{0}]_{3 \times 6} & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}_{7 \times 13} \quad (4.5)$$

where the numerical subscripts on  $\mathbf{C}^{pls}$  imply use of the entire matrix ( $3 \times 3$ ) or a scalar element from the corresponding row and column (e.g., 1, 2).

Small errors in the knowledge of gravity (much greater than those typical for modern navigation systems) should not in general invalidate this approach. Consider a five degree error in the direction of gravity (equivalent to a vertical bias of less than 0.004 g, and a horizontal bias of less than 0.09 g) or comparable errors in knowledge of gravity magnitude. With reference to the scenario given in Sect. 4.2.1, we would expect this error magnitude to have negligible effects on scenarios that involve pose estimates implying significant differences in acceleration *magnitude* (angle of attack), since small angle of attack differences for tactical aircraft quickly lead to acceleration differences much larger than one g.

Where pose estimates lead to differences in acceleration *orientation* only, the effect of gravity error could be more significant. For example, where the error in gravity direction is comparable to the difference in lift orientation between correct and incorrect pose estimates,

and the incorrect pose estimate (target) appears to be more properly oriented with respect to the (incorrect) direction of gravity than the correct pose estimate (target), this situation could lead to incorrect identifications – the incorrect target would “fly” with lower residual errors than the correct one. In any case, with an imaging sensor and knowledge of range as in the scenario of Sect. 4.2.1, differences in physical dimensions for typical targets of interest should lead to pose estimate differences in general much larger than typical errors in the knowledge of the direction of gravity.

**4.2.4 Filter Tuning.** No great deal of filter tuning was required or performed to demonstrate the behavior of interest in the proposed application. Dynamics driving (pseudo) noise ( $\mathbf{Q}$ ), Gauss-Markov state correlation times ( $\tau$ ), and measurement noise parameters ( $\mathbf{R}$ ) are shown in Table 4.1.

Most initial variances are left large (relative to the square of typical values) for states assumed “constant but unknown” to facilitate the filter converging to reasonably good values. Note that the variance for  $\alpha_b$  (the “trim” component of  $\alpha$ ) is not large, implying that it is reasonably well known. This is significant, as we shall see later.

Correlation times for Gauss-Markov processes were set based on reasonable expectations for the time required for flight control systems to null fluctuations in these states. Dynamics driving noise strength values for these Gauss-Markov angular states were then set using the procedures discussed in Sect. 2.3.2.1 and [153:178] (as for the Gauss-Markov acceleration states in the Singer target model), with reasonable assumptions for expected angular standard deviations. Dynamics driving noise values for the assumed-constant angular states are set to arbitrary small values.

Measurement corruptions in this scenario are assumed to be white. The measurement sampling interval is taken as 0.1 seconds, consistent with dedicated tracking, as opposed to track-while-scan operation. The following paragraphs consider position, velocity, and aspect angle measurements in turn.

Position measurements treat the target as a point, adding independently generated Gaussian noise samples of standard deviation 100 feet to each axis in the *true* line-of-sight frame. The filter measurement model assumes Gaussian noise of 100 foot standard

Filter Tuning Parameter	Truth Model	Filter Model
$P_{p_i/i_N,0}$	N/A	$100^2 \text{feet}^2$
$P_{p_i/i_E,0}$	N/A	$100^2 \text{feet}^2$
$P_{p_i/i_D,0}$	N/A	$100^2 \text{feet}^2$
$P_{v_i/i_N,0}$	N/A	$100^2 (\text{feet/sec})^2$
$P_{v_i/i_E,0}$	N/A	$100^2 (\text{feet/sec})^2$
$P_{v_i/i_D,0}$	N/A	$100^2 (\text{feet/sec})^2$
$P_{\alpha_b,0}$	N/A	$0.001 \text{radians}^2$
$P_{\alpha_c,0}$	N/A	$0.10 \text{radians}^2$
$P_{\alpha_{gm},0}$	N/A	$0.001 \text{radians}^2$
$P_{\rho_c,0}$	N/A	$1.0 \text{radians}^2$
$P_{\rho_{gm},0}$	N/A	$0.001 \text{radians}^2$
$P_{\beta_{gm},0}$	N/A	$0.001 \text{radians}^2$
$P_{k_{aero},0}$	N/A	1.0 (non-dimensional)
$Q_{\alpha_b}$	N/A	$1.0 \text{deg}^2 \text{sec}$
$Q_{\alpha_c}$	N/A	$1.0 \text{deg}^2 \text{sec}$
$Q_{\alpha_{gm}}$	N/A	$1.0 \text{deg}^2 \text{sec}$
$\tau_{\alpha_{gm}}$	N/A	1.0 sec
$Q_{\rho_c}$	N/A	$1.0 \text{deg}^2 \text{sec}$
$Q_{\rho_{gm}}$	N/A	$1.0 \text{deg}^2 \text{sec}$
$\tau_{\rho_{gm}}$	N/A	1.0 sec
$Q_{\beta_{gm}}$	N/A	$1.0 \text{deg}^2 \text{sec}$
$\tau_{\beta_{gm}}$	N/A	0.5 sec
$Q_{k_{aero}}$	N/A	1.0 sec
$R_{p_i/a_N}$	$100^2 \text{feet}^2$	$100^2 \text{feet}^2$
$R_{p_i/a_A}$	$100^2 \text{feet}^2$	$100^2 \text{feet}^2$
$R_{p_i/a_E}$	$100^2 \text{feet}^2$	$100^2 \text{feet}^2$
$R_{v_i/a_N}$	$60^2$ or $100^2 (\text{feet/sec})^2$	$60^2$ or $100^2 (\text{feet/sec})^2$
$R_{\alpha_m}$	$9 \text{deg}^2$	$4 \text{deg}^2$
$R_{\rho_m}$	$9 \text{deg}^2$	$4 \text{deg}^2$
$R_{\beta_m}$	$9 \text{deg}^2$	$4 \text{deg}^2$

Table 4.1. Filter Tuning Parameters

deviation relative to the *predicted* line-of-sight frame, as defined above. For spherical error ellipsoids as employed here, the difference between orientations of the true and predicted line-of-sight frames is not important.

At 100,000 feet range, this position accuracy equates to an angular accuracy standard deviation of one milliradian, which is considered to be state of the art for an aircraft radar system in a dedicated tracking mode, but very conservative with respect to the performance of state-of-the-art military land- or ship-based systems [15, 16, 11]. Likewise, assumption of 100 foot standard deviation for range measurements is considered conservative. It is important to note that position bias errors (i.e., errors constant for periods of several seconds or more) will have little effect on the proposed algorithms. "White" and otherwise rapidly changing errors, however, are observed by the filter as target accelerations, and for aircraft targets at least, imply large variations in aspect angle.

Doppler velocity measurements are similarly defined from the target-sensor relative velocity along the true boresight, but are treated by the filter as though along the predicted boresight. Both true and filter doppler velocity standard deviations are 60 feet per second in Test Scenarios 1A and 1B (defined in the next section). This accuracy is conservative with respect to published results [48]. True and filter doppler velocity standard deviations were increased to 100 feet per second (i.e., approximately twice the accuracy of results in [48]) to reduce considerably, but not eliminate, the impact of doppler information for Scenario 2 – this is the only tuning difference between the three scenarios.

Aspect angle measurements are corrupted in all cases with white noise of standard deviation three degrees, while the filter assumes two degrees. This was an arbitrary choice, rather smaller than the five degree standard deviations used as nominals by [120], but reflecting an ultimate intent to obtain (through new methods discussed in the previous chapter, and demonstrated in the following chapter) smoother pose estimates than have been available in the past. The increase in true over filter-modelled noise is simply intended to stress the filter somewhat. Bias error values are discussed in the following section, since they were scenario-specific.

### 4.3 Results and Discussion

The proposed filter was implemented using the "Multimode Simulation for Optimal Filter Evaluation" (MSOFE) software [46], as developed by Mr. Stan Musick of Wright Laboratory (WL/AAAS) *et al.* The evaluation was conducted primarily by feeding the filter with artificially biased and white noise-corrupted aspect angle estimates. These corruptions are assumed to have arisen, for example, in the manner as described in Sect. 4.2.1 – this demonstration does not provide for an explicit simulation of the pose estimation process. All results are for 20-run Monte Carlo sets with statistics as given in the previous section.

These results will show decisively that operating a kinematic/aspect-angle filter with inappropriate model assumptions can lead quickly to large residuals or deviations from expected state values. The comparison of these residuals and states of interest with allowable covariance values using Bayes' Rule and multiple model algorithms as described in Sect. 2.3.1.3 and [154:129–136] is a powerful approach for indicating the correct parameter set, or target class, that generated the observed combination of kinematic and feature observable behavior. It will be clear, moreover, that in many cases, a simple thresholding algorithm will quickly eliminate some target classes from consideration. Another point to note is that certain scalar residual elements and states may be more indicative of incorrect target class associations than others – observing behavior in these values *individually* may be more practical than observing, say, the ensemble log likelihood of all residuals and states of interest, where differences can become "blurred" in the fluctuations common to all models.

Most of the the plots showing the natural log of the residual/state likelihood for the various scenarios are computed according to Eqn. (4.6). Note that this likelihood is *not* a moving window sum or a sum over an ever-increasing time period, rather the individual log likelihood value at the particular time. The intent here is to show that, in the mean, significant residual differences can be developed at any one time. Two sets of plots will show how likelihood separation increases with the use of a moving window sum, over one second (ten measurement events, in those cases).

Also, note that Eqn. (4.6) treats the residual vector elements as independent from each other at any given event, which is not strictly conventional – generally, the log likelihood is calculated using the weighted inner product of the residual vector with itself at time  $t_i^-$  (prior to any updates), where the weighting matrix is the inverse of the (classical) residual covariance matrix  $\mathbf{A}(t_i) = [\mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}^T(t_i) + \mathbf{R}(t_i)]$  (see Sect. 2.3.1.3). However, MSOFE performs sequential scalar measurement updates, and the residual values and corresponding variances are made available sequentially as scalars. Although not formally proven for this effort, it is evident (as implied by Kailath [118]) that residual covariance-weighted distance in the “innovations space” is an information measure, and, under *linear* filtering assumptions, we should not expect it to change with method or order of processing. For the case of two-dimensional measurements, equivalence is easily shown between the quadratically-weighted squared distance obtained from vector residuals as noted above, and the squared distance obtained from sequential scalar updates using the following equation. Thus, the likelihood of interest is taken as:

$$L_1(t_i) = -0.5 \left\{ \left[ \sum_{m=1}^7 \frac{r_m^2(t_i)}{\sigma_m^2(t_i)} \right] + \frac{[1.0 - \hat{k}_{aero}(t_i^+)]^2}{P_{k_{aero}}(t_i^+)} \right\} \quad (4.6)$$

where:

$L_1(t_i)$  = the (natural) log likelihood of the observed residuals and state for one (hence subscript “1”) measurement event at time  $t_i$ , less the bias factor associated with the log likelihood of a Gaussian probability density.

$r_m^2(t_i)$  = the residual (squared) for the m-th measurement at time  $t_i$ .

$\sigma_m^2(t_i)$  = the filter-computed variance for the residual of the m-th measurement at time  $t_i$ .

$P_{k_{aero}}(t_i^+)$  = the filter-computed estimate of the variance in the aerodynamic scale factor  $k_{aero}$  (the 13th state) following measurement update at time  $t_i$ .

and other variables are as defined above.

We note in passing that for nonlinear filtering, order of computation can be very important – in general, we often choose to process the most accurate information first,

in particular to provide the most accurate start for successive relinearizations. In the sequence of scalar updates for this effort, we process position measurements before the doppler measurement – to provide a best estimate of target direction along which to align and process the doppler velocity information.

**4.3.1 Scenarios 1A and 1B.** The following discussion refers to a situation very similar to that shown in Fig. 4.1 – a target is turning toward us (the sensor) in a horizontal plane so that the plane of the target wings is generally normal to the sensor-to-target vector. This is a scenario of great interest in air-to-air combat, due to the possibility that a hostile target will release a missile at the completion of the turn. We wish to determine the class of the unknown target engaged in this maneuver.

Each of the points of this discussion revolves around a pair of figures – the first figure showing what a properly-matched target model would yield, and the second figure showing what a particular mismatched target model would yield. The effect of the mismatched model in this case is to induce: (1) a 20 degree positive bias in the roll angle pseudo-measurement (i.e., the target is apparently rolled 20 degrees further to the right than the actual roll angle required for the given maneuver) and (2) a 20 degree positive bias in the angle of attack pseudo-measurement. The reader should note the extent to which, in each case, the mean measurement residual or state estimate remains within the bounds of the residual or state filter-computed standard deviation,  $\sigma_f$  – the symmetrically paired curves represent zero (the filter-anticipated residual mean)  $\pm$  one  $\sigma_f$ .

Figs. 4.3 through 4.10 refer to Scenario 1A, in which the target turn rate equates to 1 g in the horizontal plane (so that the nominal roll angle is 45 degrees to the local horizontal). For Figs. 4.11 through 4.14, or Scenario 1B, we increase the horizontal turn acceleration to 4 g's, so that the nominal orientation of the plane of the wings is nearly perpendicular to the local horizontal.

Figs. 4.3 and 4.4 show the behavior of the vertical position measurement ( $\tilde{p}_{t/a_A}$ ) residuals with correct and incorrect model assumptions. The second (incorrect model) figure exhibits classical residual "failure", which may be interpreted as the filter directing its acceleration estimate in the wrong direction, and being unable to supply the lift required

to maintain the filter model near the altitude of the true target – hence, these altitudes diverge, and the divergence appears immediately in the residual. This behavior will be observed with or without the angle of attack bias. Note that the plots look essentially the same over the first two seconds because some time is required for the underlying *acceleration* contradiction to develop an obvious *position* contradiction (moreover, the same noise realizations were used for both plots).

Figs. 4.5 and 4.6 show the behavior of the state estimate for  $k_{aero}$ . As noted above, this state is expected to have a value of one. Errors are reasonably small for the unbiased angular pseudo measurements – the “wandering” effect over time is due to limited observability between this and other states (principally the angle of attack states). For biased angle of attack measurements, however,  $k_{aero}$  reflects immediate deviation from its nominal value. This result may be interpreted as the filter telling the user that it is willing to accept the observed position, velocity, and aspect angle values, but that the target must have *many* times the nominal expected mass, in order to be consistent with the observed position and velocity measurements over time. This behavior will be observed with or without the roll angle bias.

Note that the behavior of the  $k_{aero}$  estimate for the wrong model is due to the fact that the initial variance for state  $\alpha_b$ , the “trim” angle of attack value, is kept small and the initial  $\alpha_b$  value is zero. Increasing the initial variance for this state will allow it to assume significantly non-zero values, reducing the tendency for the filter to modify  $k_{aero}$  according to observed measurements. In that event,  $\alpha_b$  should be added to Eqn. (4.6) as a state of interest – looking for deviations from a nominal trim angle of attack. Alternatively, “locking”  $k_{aero}$  to a value near one and  $\alpha_b$  to values near zero will cause immediate deviations in the range rate residual, if the boresight vector is aligned along axes of motion affected by these assumptions. The key point is that, *one way or another*, where kinematics and aspect angle are as highly coupled as they are for most aircraft, incorrect modelling assumptions will betray themselves in a kinematic/aspect-angle filter.

Next, Figs. 4.7 and 4.8 contrast the behavior of the log residual/state likelihood *at each measurement event* (not a sum over several events) for the two target model choices. The somewhat lower likelihoods around the one second point in time are due to the angle



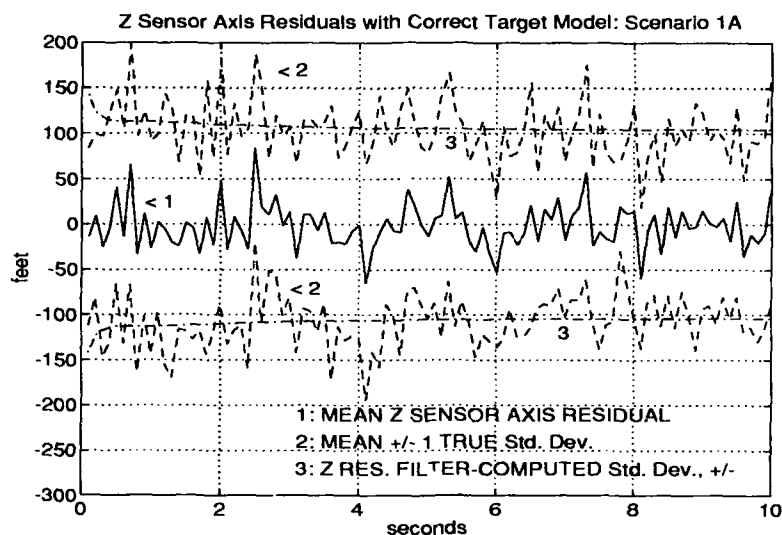


Figure 4.3. Z (Vertical) Sensor Axis Position Residuals, Correct Model, Scenario 1A

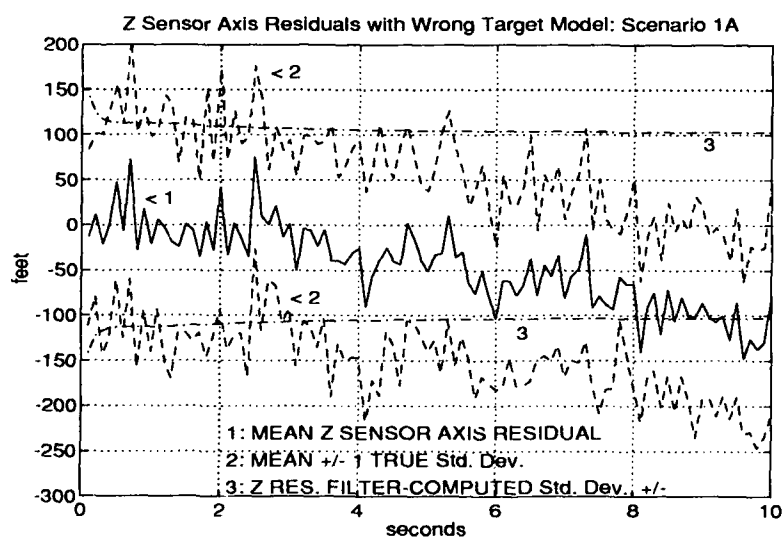


Figure 4.4. Z (Vertical) Sensor Axis Position Residuals, Wrong Model, Scenario 1A

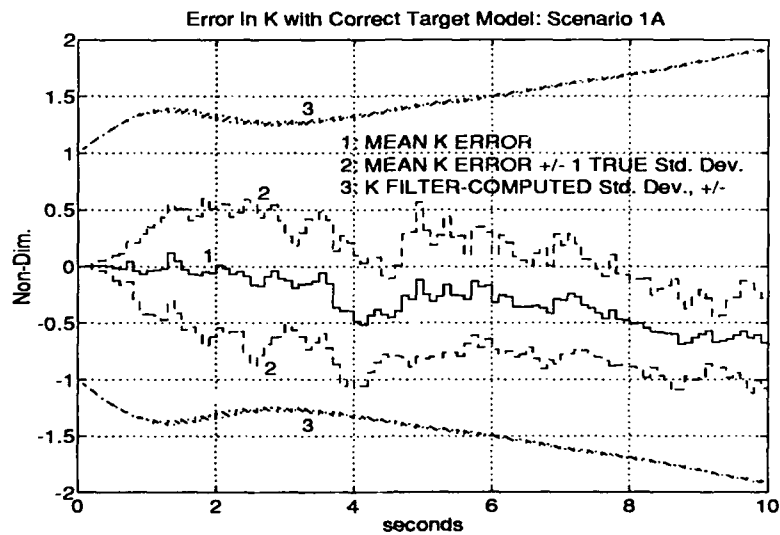


Figure 4.5.  $k_{acro}$  Error, Correct Model, Scenario 1A

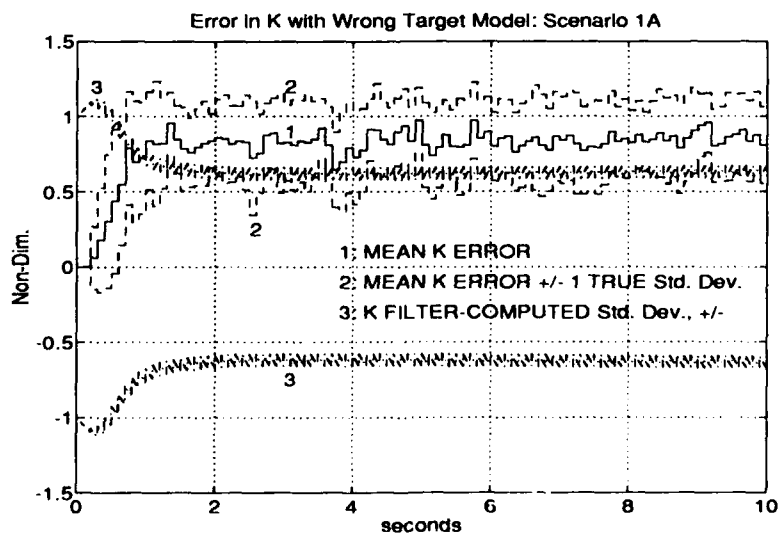


Figure 4.6.  $k_{acro}$  Error, Wrong Model, Scenario 1A

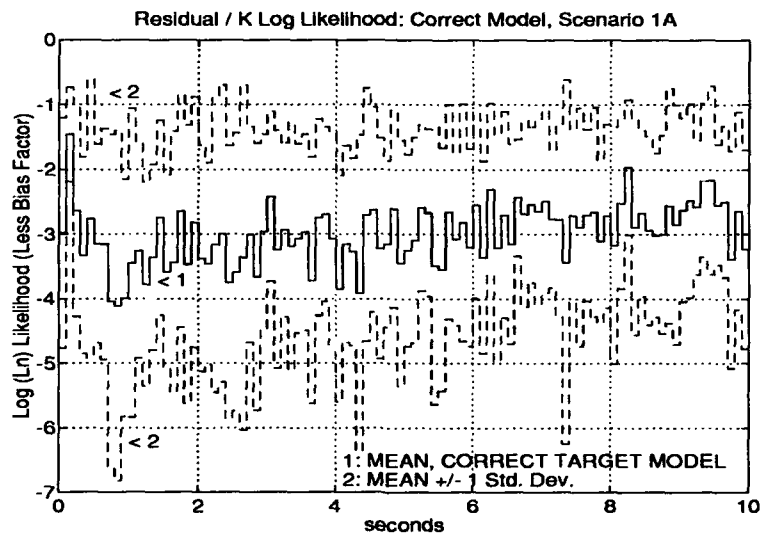


Figure 4.7. Residual / State Log Likelihoods, Correct Model, Scenario 1A

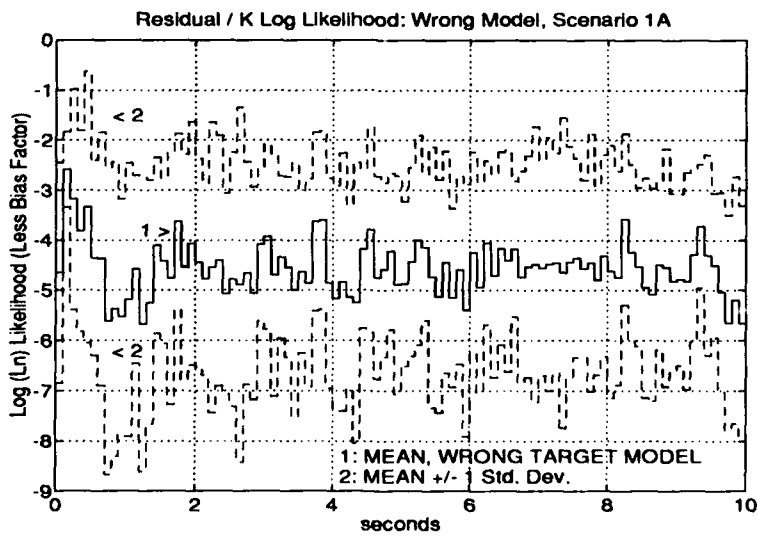


Figure 4.8. Residual / State Log Likelihoods, Wrong Model, Scenario 1A

of attack error, and the recovery occurs because of the changed or "learned" value in  $k_{aero}$ . The unreasonable resulting value of  $k_{aero}$ , however, is the major factor in the remaining difference between residual/state log likelihood for the correct and incorrect target models.

The reader should expect that using partial or growing sums of log likelihoods over time will increase the distinction between likelihoods for the two cases. The effect of summing the likelihoods over a moving window of one second (ten measurements) is illustrated in Figs. 4.9 and 4.10. Note the improved separation between correct and incorrect models for this pair of figures, by comparison with Figs. 4.7 and 4.8. Treating these likelihoods in a two-class Bayesian parameter estimation case (Eqn. (3.2)) with equal *a priori* probability  $p(\omega_j)$ , we quickly observe that since the mean likelihood for the wrong case ( $\approx 10^{-10} \exp(-45)$ ) is *many orders* of magnitude smaller than that for the correct case ( $\approx 10^{-10} \exp(-30)$ ), the *a posteriori* probability for the "correct" (unknown *a priori*) target, in the mean, is effectively one (note that the leading coefficients for the two Gaussian likelihoods are both on the order of  $1.0 \times 10^{-10}$ ).

Now we consider Scenario 1B. Figs. 4.11 and 4.12 show that higher turn rates speed the development of residual error in the vertical sensor position channel. Figs. 4.13 and 4.14 show that the  $k_{aero}$  state behavior is consistent with the earlier case (as are the residual log likelihoods, not shown here). Note that  $k_{aero}$  state error appears somewhat smaller in Scenario 1B than Scenario 1A because of, and not in spite of, the higher true acceleration level in this case. The *proportional* difference in mass implied by an increase (error) of 20 degrees in angle of attack when the required angle of attack is large (for the high-g turn) is *less* than the proportional difference in mass implied by an angle of attack error of 20 degrees when the required angle of attack (turn rate) is smaller.

It is important to note that the tuning parameters in these scenarios represent what are for many applications *very* worst case choices, both for the filter and truth model. For example, in an air-to-air scenario with a radar and an imaging sensor, even with many miles of range to target, the random component of position measurement error normal to the line-of-sight could be considerably better than the 100 foot one sigma errors used here, if the imaging sensor aids in the pointing and tracking process. This will reduce the random noise in these results considerably, while preserving the trends shown here. Also

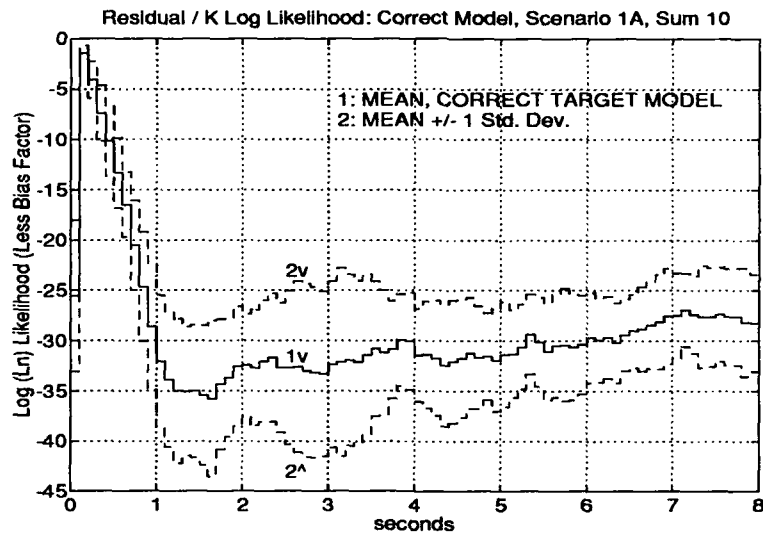


Figure 4.9. Residual / State Log Likelihoods, Correct Model, Scenario 1A, Moving Window Sum (one second, ten measurements)

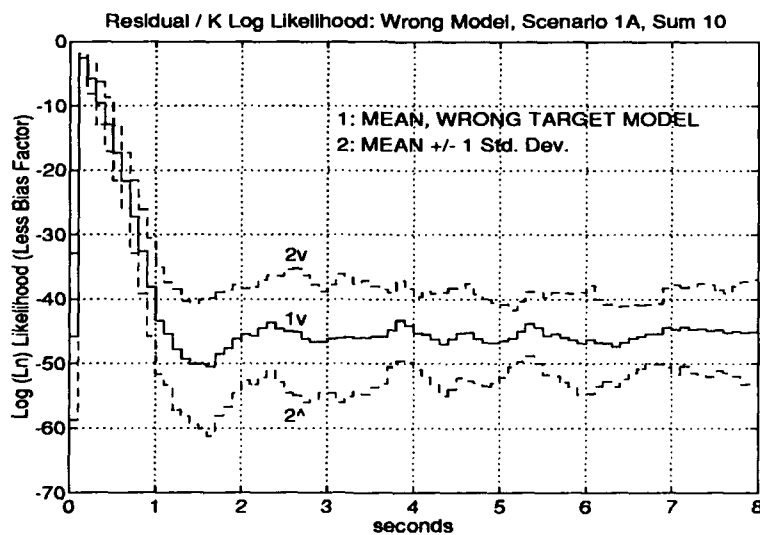


Figure 4.10. Residual / State Log Likelihoods, Wrong Model, Scenario 1A, Moving Window Sum (one second, ten measurements)

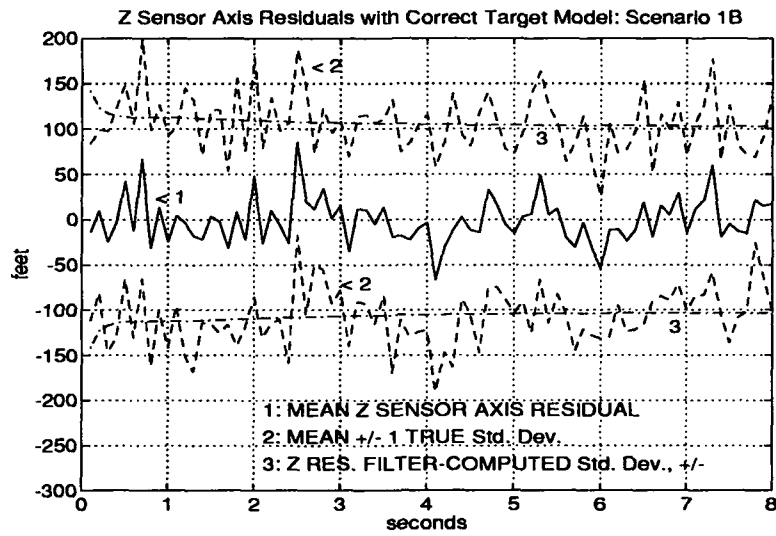


Figure 4.11. Z (Vertical) Sensor Axis Position Residuals, Correct Model, Scenario 1B

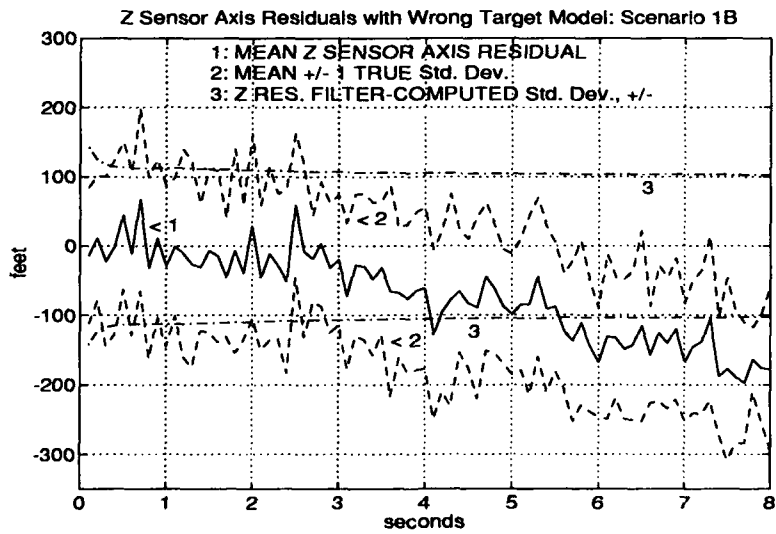


Figure 4.12. Z (Vertical) Sensor Axis Position Residuals, Wrong Model, Scenario 1B

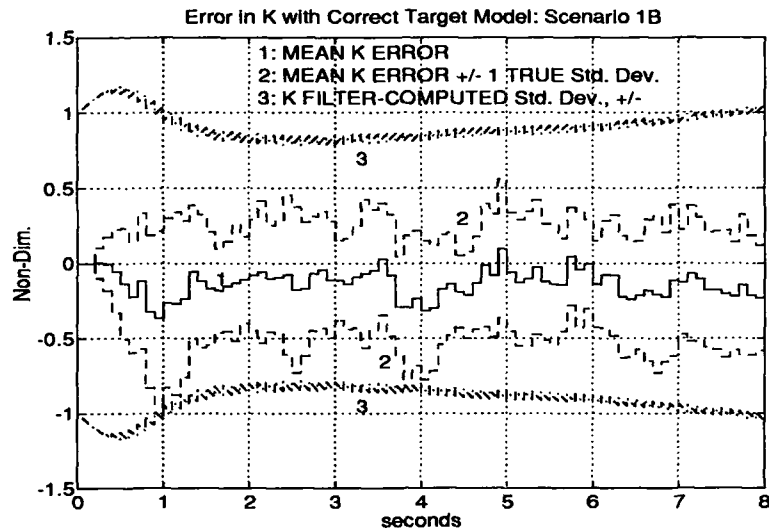


Figure 4.13.  $k_{aero}$  Error, Correct Model, Scenario 1B

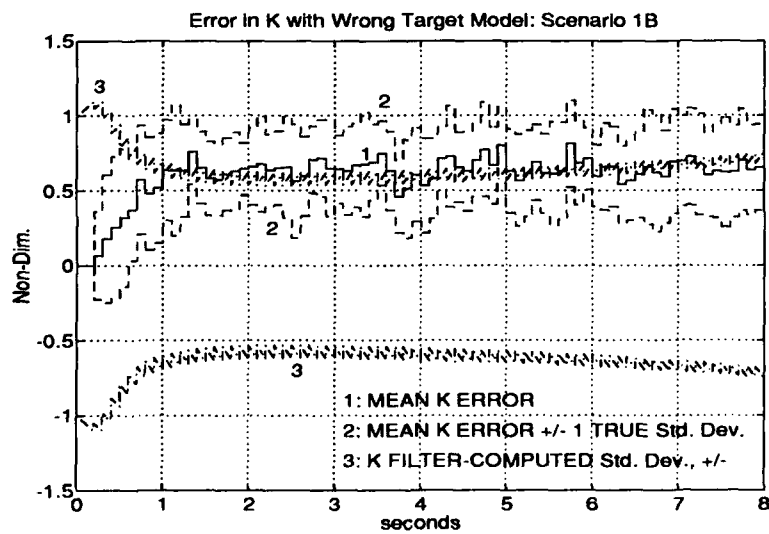


Figure 4.14.  $k_{aero}$  Error, Wrong Model, Scenario 1B

recall that these results are for a case in which the aspect angle pseudo-measurements are rather noisier than assumed by the filter – this means that the filter tends not to suppress the noise as well as it could if it were “told” that the noise level is higher. Thus, tuning could improve these results somewhat, but the intent here is to show that precise tuning is not required to demonstrate major differences between classes in likely scenarios.

*4.3.2 Scenario 2.* This scenario is very simple – the target is 20 miles out, 10,000 feet higher than the sensor, and flying at 800 feet per second straight and level on a bearing directly toward the sensor. This scenario is unique among the ones addressed in this research in that the aircraft isn’t turning – in this case, the combination of conditions required merely for level flight will quickly indicate an incorrect target class. Recall from Sect. 4.2.4 that the doppler velocity information quality is degraded here over that in the previous case, in particular to show that high quality doppler isn’t required – the activity of interest here (conflict in the vertical acceleration estimate) occurs *normal* to the line-of-sight.

As before, we have two candidate target classes – the correct one, and an incorrect one. Again, the pose estimator gives an aspect angle measurement which equates to a 20 degree positive bias error in angle of attack. This could occur for this trajectory, for example, if a high range resolution radar algorithm were “allowed” to find the best pose match from a short true target to a long library model – generally looking beyond the 10 or 20-degree square extent of aspect angle window to which searches would normally be restricted (bearing in mind, as noted in Sect. 3.8.4, that small aspect angle search windows based on *real time* tracking for aircraft targets are highly unrealistic in any case). Figs. 4.15 and 4.16 show that the incorrect model association betrays itself quickly in the  $k_{acro}$  state as before. In general, it has been noted that any angle of attack bias with an absolute value of ten or more degrees causes comparable results for aircraft of the F-4 / MIG-21 performance range.

Figs. 4.17 and 4.18 (likelihoods at each measurement) and Figs. 4.19 and 4.20 (summing likelihoods over a moving window of one second or ten measurements) show that residual differences are somewhat more pronounced between correct and incorrect models



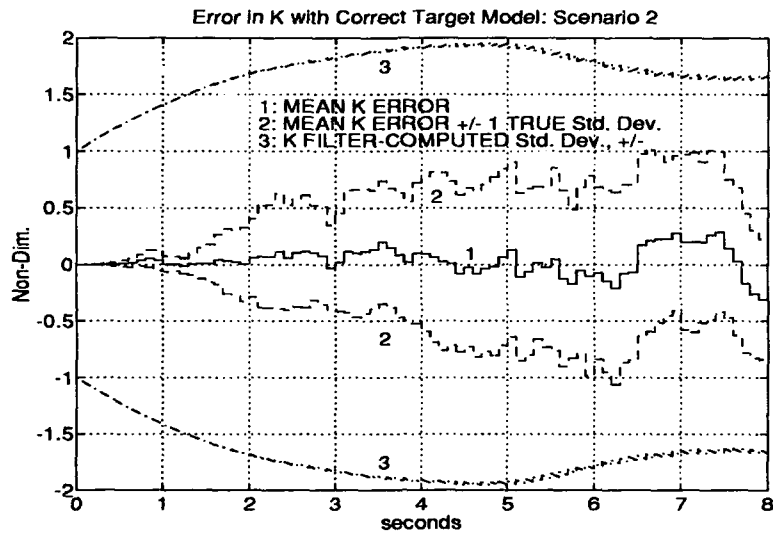


Figure 4.15.  $k_{aero}$  Error, Correct Model, Scenario 2

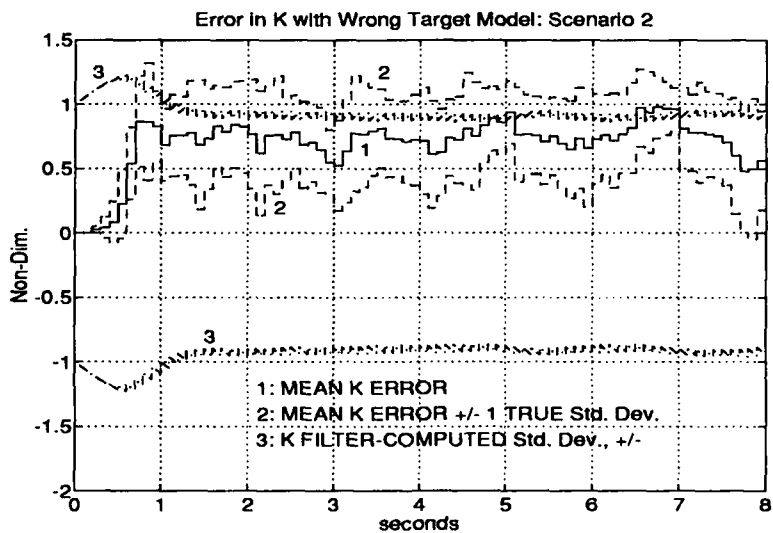


Figure 4.16.  $k_{aero}$  Error, Wrong Model, Scenario 2

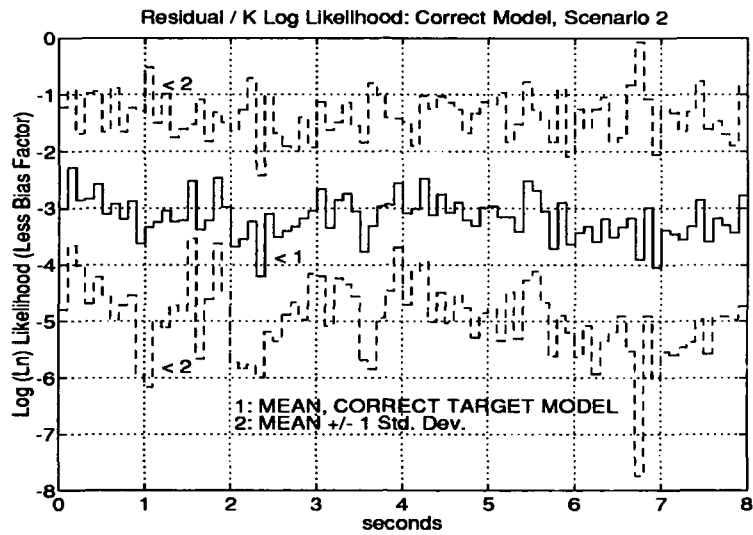


Figure 4.17. Residual / State Log Likelihoods, Correct Model, Scenario 2

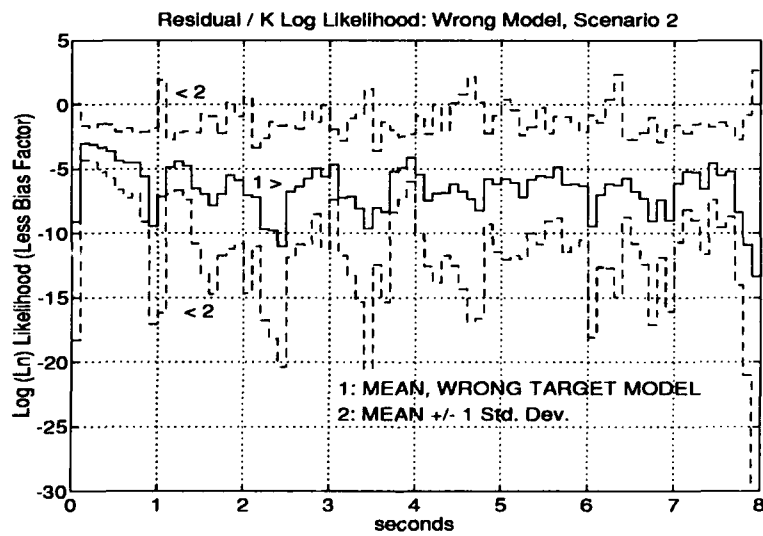


Figure 4.18. Residual / State Log Likelihoods, Wrong Model, Scenario 2

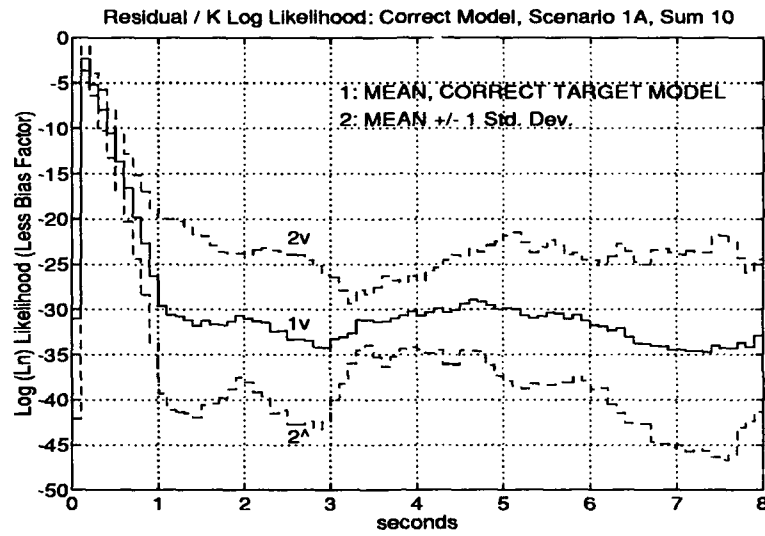


Figure 4.19. Residual / State Log Likelihoods, Correct Model, Scenario 2, Moving Window Sum (one second, ten measurements)

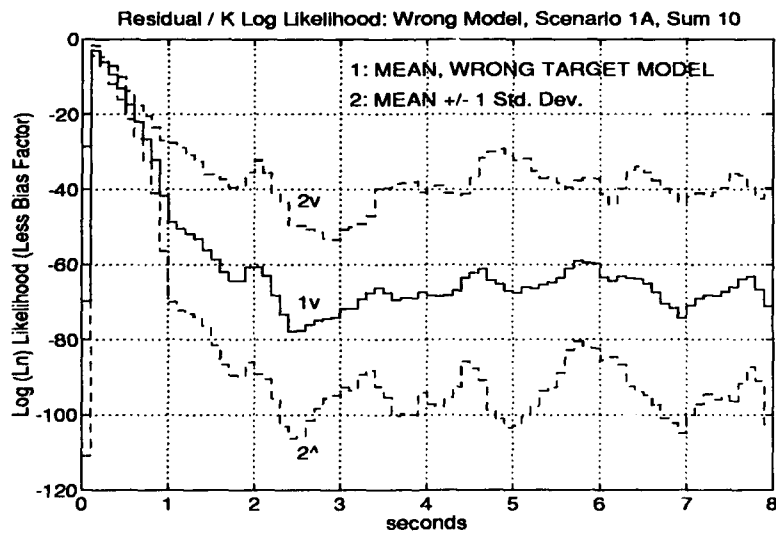


Figure 4.20. Residual / State Log Likelihoods, Wrong Model, Scenario 2, Moving Window Sum (one second, ten measurements)

than in the previous scenarios. This is due both to the poorer doppler quality and different trajectory here, compared with that in the previous scenarios. Restoring the better doppler quality will improve these residuals somewhat, but *artificially* providing the filter with "vertical" velocity measurements (i.e., target velocity measurements along the axis of the vertical motion which would tend to be induced by a large angle of attack in this case) is observed to restore the residual error to slightly smaller than that observed in Fig. 4.8 – as expected, since Scenario 2 does not have a roll angle measurement bias. The point here is that high quality measurements in a particular state space dimension can retard the development of high error residuals in that dimension, even when the model parameters governing state behavior in that dimension are in error.

In other words, high quality measurements can force the state dynamics model to follow observed behavior, despite the model's inclinations to the contrary. In that case, residual behavior *per se* may not be the best indicator of an improper parameter set choice – hence our interest to consider indicators such as the behavior of states like  $k_{aero}$ .

Summarizing, the key point in this scenario is that the translational/rotational coupling for conventional airplanes is so severe that relatively small pose estimate differences can reveal incorrect target class choices dramatically *even in straight, level flight*. The conventional approach to target recognition by comparing distances in *feature observable spaces* for various target class choices, without considering dynamic implications, is not nearly as likely to yield an output that so forcefully indicates incorrect choices. This is simply further confirmation of the imperative to consider the *joint likelihood* of all observable events in making object recognition decisions.

#### 4.4 Summary

The results of this chapter provide clear confirmation of claims made in Chapter III: that kinematic/aspect filters driven by correct and incorrect pose estimates under nonrestrictive conditions can exhibit distinctly different residual and state likelihoods. These likelihoods can be used in a Bayesian multiple model residual analysis parameter estimator for target recognition – *without* explicitly comparing observed and predicted feature observable measurements for correct and incorrect classes.

This application is only one in a class of techniques that can be used for dynamic object recognition whenever (1) two or more disjoint subsets of estimator state variables are highly coupled by parameters that differ between object classes, (2) when measurements or pseudo-measurements are available that are functions of state variables from the disjoint subsets, and (3) the objects are observed under conditions that exploit state variable coupling. These techniques will consider the *joint likelihood* of measured and pseudo-measured events, conditioned on (1) state coupling rules associated with each candidate parameter set, and (2) prior measurements and pseudo-measurements – including, implicitly, the feature observable measurements from which pseudo-measurements were derived.

These results confirm a significant theoretical and practical contribution to the field of dynamic object recognition, extending the proposals of Therrien [211] to cases in which predictor/corrector methods are not suitable for all measurement quantities, but in which low ambiguity mappings exist between such ill-defined measurement quantities and one or more uniquely coupled estimator states. The results also demonstrate in particular a new and powerful approach for aircraft recognition.

In the following chapter, we demonstrate methods for explicitly comparing feature observable measurements to feature observable libraries for known classes, under conditions where predictor/corrector methods may be infeasible. In particular, we will be concerned with combinations of rapid, unpredictable state space transitions and nonlinear measurement functions that combine to create “high frequency” measurement variations – a set of conditions that applies in general for high range resolution (HRR) radar target recognition.

## *V. Dynamic Programming Sequence Comparison for Target Recognition*

### *5.1 Introduction*

This chapter will expand upon the discussion in Chapters I and III to demonstrate the proposed concepts of dynamic programming-based sequence comparison – “motion warping” – for sensor fusion and object recognition. This is the second of three steps for considering the joint likelihood of feature observations and kinematic measurements, as discussed in Sect. 3.1. The objective here is to define and investigate a class of algorithms that combines feature observable information while obeying feasible or observed kinematic constraints for the objects of interest. We will find that restricting feature observable associations to kinematically likely sets achieves the result predicted in Sect. 2.6.2 – that is, reducing the output likelihood value for incorrect associations, without substantially affecting the output likelihood values for correct associations.

As in the previous chapters, the primary interest here is recognition of aircraft targets, and the terminology will reflect that interest. Again, however, these basic approaches are suitable for application to many different recognition tasks.

### *5.2 Concept Overview*

The process of motion warping as demonstrated in this chapter is briefly summarized in the following steps, to provide a basis for understanding the more technical discussion in the sections that follow.

*5.2.1 Tracking the Target.* This process can use in general any of the recognized mathematical models for target tracking. The research described in this chapter uses the standard extended Kalman filter / Singer model kinematic tracking approach as described in Sect. 2.3.2.1, followed by optimal fixed lag smoothing and polynomial curve fitting to develop trajectory information suitable for estimating acceleration of an aircraft target in a turn. Smoothing and polynomial curve fitting are covered in detail in App. C.6.

*5.2.2 Defining Likely Aspect Angle Regions and Paths.* Given an estimate for the target position, velocity, and acceleration over some time period of interest, for most

target classes of interest we can develop estimates for the most likely, or nominal aspect angle path traced by the target-sensor vector on the hypothetical aspect angle sphere, as discussed in Sect. 2.3.3.1. Knowledge of statistics for the error in target and ownship state will allow us to define aspect angle error bounds around the nominal aspect angle path.

*5.2.3 Matching Feature Observations to Aspect Angles.* At this point, we have defined aspect angle regions on target models of interest and nominal paths through those regions, i.e., paths kinematically likely to have been traced as measured feature observable sequences were generated. Target sensor signature modeling tools are used to provide signature libraries corresponding to required models and aspect angles. From one of these models, we can select a feature observable sequence which will be corrupted with noise to produce the "true" sequence. Now we use any of several dynamic programming sequence comparison techniques (as discussed in Sect. 2.4) or other methods to match feature observables with likely aspect angle states for each feasible target model.

*5.2.4 Evaluating Performance.* We will see that the performance results of dynamic programming sequence comparison can be expressed in terms of generalized ambiguity functions, as well as in more conventional terms such as probability of correct or incorrect recognition.

### *5.3 Generation of Simulated Target Signatures*

*5.3.1 Introduction.* As the author began to develop dynamic programming (DP)-based sequence comparison for moving object recognition, Mr. Michael Bryant and Mr. Rick Mitchell of the Target Recognition Technology Branch (WL/AARA) at the USAF's Wright Laboratory (USAF sponsor for this effort) requested that the technique be applied to aircraft recognition using high-range resolution (HRR) radar. This choice of sensor signature domain was influenced primarily by the availability of computer models for signature generation. The fact that dynamic time warping (DTW) for speech is notably robust with regard to choices of feature space and metric was also reassuring.

Thus, HRR radar target signature libraries were a fundamental requirement for pursuing this research. Achieving high fidelity with respect to real-world signatures from actual targets of interest, however, was not a requirement. Recall that the intent here is not to establish performance absolutes, but simply to demonstrate *reduced ambiguity* in object recognition using sequence comparison techniques, for some choice of signature and metric, relative to more conventional approaches. That being said, the effort would clearly be more interesting if some connection is made between the research targets and real targets, and research metrics and real metrics. The chosen approach to radar signature generation reflects this compromise.

Some measured signature data was available [166, 20], but the author made only limited use of this data for several reasons. First, most measured data is derived from classified research, and, even where the data itself is unclassified, much information of interest about the data remains classified (particularly associations between the data and the actual target classes). This dissertation is required to be unclassified.

Second, the available measured data is limited with respect to aspect angles and polarizations (*polarization*, as discussed in Sect. 2.2.3, refers to the orientation of the incident radar waveform in roll angle around the sensor-to-target vector, with respect to the target). Most available aircraft signature data tends to be taken from aspect angles near the plane of the target's wings. For our purposes, this region tends to be of little interest, since a turning aircraft target at any altitude reasonably near that of the sensor quickly exhibits aspect angles that are far from the plane of the wings. Use of measured data, then, would have severely limited the possible target trajectories.

Third, measured data is available for only a few distinct target classes. For our purposes, we will desire to define *pseudo*-targets, in some sense between real target classes of interest. By definition, measured data will not provide signatures for these targets. Other problems with measured data from recent tests aside from those mentioned here are discussed at length in [20] (classified secret).

For these reasons, the author and the research sponsors (WL/AARA) determined that synthetic signature generation was the proper approach for this research. The ini-



tial choice was the Syracuse Research Corporation radar signature simulation "SRCRCS" developed under contract to the USAF's Phillips Lab at Rome, N.Y. [53]. This tool was familiar both to AFIT and WL/AARA, and was hosted on computer systems at both locations.

SRCRCS has two significant shortcomings that limited its utility, however. First, it is written to run on a VMS operating system, and without further effort would not be compatible on UNIX-based Sun SPARC workstations, where the bulk of this research was to be accomplished. Second, it is deterministic – that is, the particular target model, center frequency, bandwidth, aspect angle, and polarization value completely define the signature. As we have seen, for real targets, even when these factors are constant, the signature will vary randomly over time, due to body vibrations and other factors.

The first of these problems was overcome by using the MIT/Lincoln Laboratory RCSToolLLBox radar signature simulation tool [37]. Other tools, some of which have much higher fidelity than RCSToolLLBox, were judged to be less mature, more complex, or more computationally demanding than the objectives and timelines for this research required. The second problem was overcome by adding noise of realistic statistics to deterministic signatures. The following sections describe these actions in detail.

*5.3.2 Signature Simulators: RCSToolLLBox and Others.* RCSToolLLBox [37] is a derivative of SRCRCS, developed by Dr. E.C. Burt (formerly of Syracuse Research Corporation) and others at MIT Lincoln Laboratory. RCSToolLLBox is designed to run on UNIX operating systems – primarily Silicon Graphics systems, which offer excellent graphic visualization capabilities. Dr. Burt provided the author with a version optimized to operate on Sun SPARC workstations, albeit without full graphics capability.

Other than operating system, from this user's perspective the only significant difference between SRCRCS and RCSToolLLBox is that RCSToolLLBox models only perfect conductors, rather than the full range of conductivities modeled by SRCRCS. In both cases, targets are modelled as three-dimensional combinations of polyellipsoids (heuristically, "warped cylinders"), triangular plates, and point scatterers that physically resemble desired targets to any desired degree. The radar scattering process is modelled using ge-

ometric optics and physical diffraction assumptions. A particularly useful fact about the relationship between SRCRCS and RCSToolBox is that the detailed but user-friendly operators' manuals for SRCRCS [53] apply in many cases verbatim to RCSToolBox.

Multiple bounces by incident radiation are not modelled by SRCRCS and RCSToolBox. Therefore, their output signatures do not exhibit in particular the *cavity* or *corner reflector* effects that are typical of real signatures from aircraft and other tactical targets. The scattering assumptions produce highly specular, or mirror-like signatures – that is, over most aspect angles, we receive generally less return radiation than a real target would return, but a few aspect angles return substantially more. SRCRCS and RCSToolBox-derived HRR radar signatures do exhibit the relative motion and aspect angle variation of returns from major scatterers that are characteristic of actual HRR radar returns, but their principal attraction is computational economy rather than precision.

The current primary modelling alternative to SRCRCS and RCSToolBox in use at the Wright Laboratory Target Recognition Technology Branch (WL/AARA) is a *ray casting* model called XPATCH, which uses geometric optics assumptions, allows for multiple bounces, and employs highly detailed target models to provide target signature estimates that correspond to observed results much more closely than SRCRCS and RCSToolBox. WL/AARA expects XPATCH to play a key role in generation of signature libraries for on-line target recognition with HRR radar. On a SPARC terminal, however, generating one XPATCH signature (i.e., for a given target model, aspect angle, and polarization value) may require several hours, whereas an RCSToolBox signature for models of the complexity level used in this research (shown below) takes about 30 seconds. Since a typical analysis run requires about 2000 signatures for each of several models, generation of XPATCH signatures for this research was considered to be out of the question. Had adequate XPATCH signature libraries been available at the start of this research, however, they would have been used, and extensions to this research should certainly employ XPATCH or comparable signature data if available.

As of this writing, additional signature simulation tools have become available locally that show great promise for future effort in multisensor fusion. Chief among these is the "BRL-CAD" Package [65], or Ballistics Research Laboratory Computer Automated Design

Package, available from the U.S. Army Ballistics Research Laboratory (BRL) at Aberdeen Proving Ground, Maryland. This package includes exceptionally detailed target modeling capability, and has been augmented by a set of radar signature calculation codes. These include the Simulated Radar IMagery, or SRIM code by the Environmental Research Institute of Michigan (ERIM), the SARSIM codes by Northrop, and the TRACK code by Georgia Tech Research Institute (GTRI). The first code is a ray casting signature generator apparently similar to that in XPATCH, while the latter two codes are apparently more comparable to RCSToolLLBox. The reader is referred to Vol. I of [65], or equivalently to [64] for a good discussion of these models and the radar modelling tradeoffs in BRLCAD, which illuminate equally well the comparison of SRCRCS/RCSToolLLBox and XPATCH above.

Most notably in contrast to RCSToolLLBox and XPATCH, however, BRLCAD models the target image in visual and infrared spectra – “multispectral” capability is essential in multisensor fusion research. BRL-CAD was not on-line at AFIT at the start of this research, but is at present.

*5.3.3 RCSToolLLBox Models and Signatures.* SRCRCS and RCSToolLLBox target models are defined in an ascii text file format called a “SCAMP” file. A SCAMP file defines the collections of polyellipsoids, triangular plates, and point scatterers that define a target in three-space, and provides supplemental information regarding the scattering properties of these objects. Modelling procedures for SRCRCS and RCSToolLLBox are described in detail in [53]. One particular point that requires emphasis is that care must be taken to ensure that SCAMP model surfaces do not intersect, since intersections tend to confuse the shadowing calculation algorithm. Fig. 5.1 shows the SRCRCS/RCSToolLLBox target body coordinate frame convention, which differs from the airframe convention used in the flight control community (to be shown in Fig. 5.23).

SRCRCS and the full-graphics versions of RCSToolLLBox can render SCAMP file contents as a visual image from any aspect – if desired, causing surfaces not visible to the radar to be graphically invisible also. Figs. 5.2 through 5.7 show typical SCAMP models used early in this research, as rendered on SRCRCS (recall that the version of

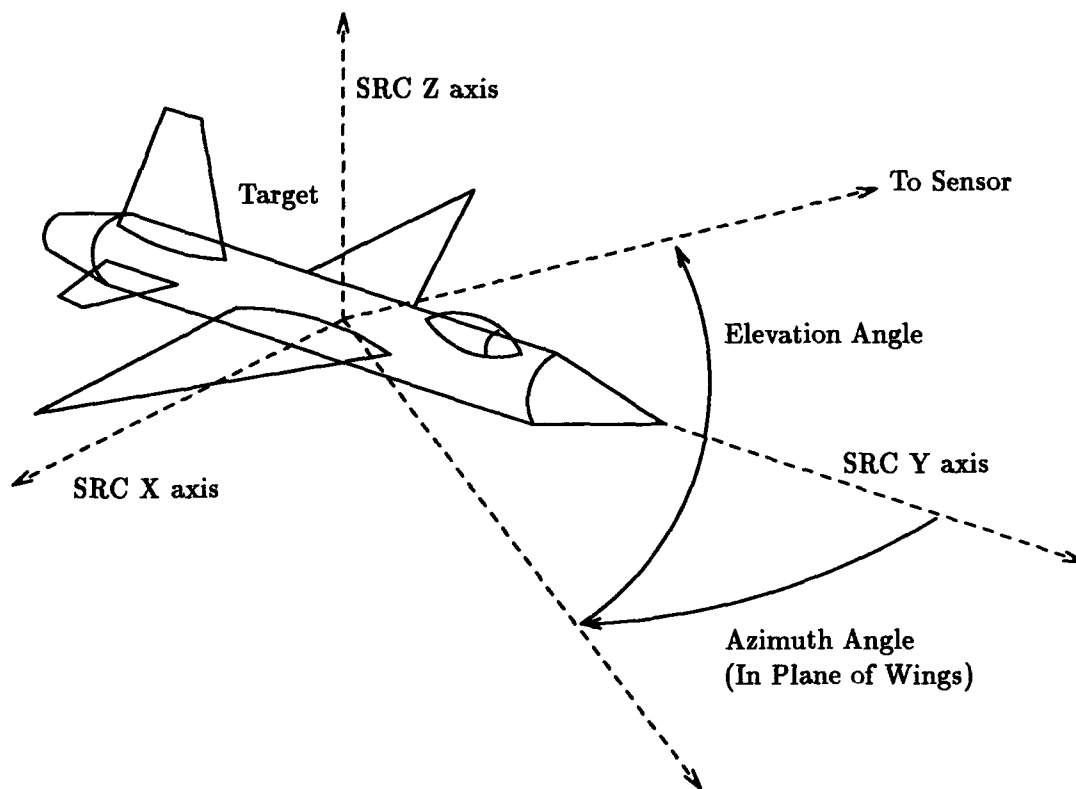


Figure 5.1. The SRCRCS/RCSToolBox Radar Target Coordinate Frame

RCSToolLLBox used in this research does not provide graphic displays). Note that the asterisks in these figures represent point scatterers, and that some surfaces have been declared as "one-sided" scatterers, and are not visible in this representation.

Figs. 5.2 through 5.7 were attempts to make "faithful" representations of typical targets of interest. Analysis of target recognition performance algorithms using these models, however, will show that for the trajectories, signature metric, and noise levels employed, all algorithms selected the correct target 100% of the time (although sequence comparison methods did so with a higher confidence level, or lower ambiguity, as observed using the generalized ambiguity function). To create more *ambiguous targets*, then, the author was ultimately driven to augment the target models with additional point scatterers. Fig. 5.8 shows such a MIG-21 so augmented. Also, where overall target size appeared to be a key discriminant, the author scaled targets relative to one other to reduce significant size differences. Only with such targets, particular trajectories, and high noise levels was it possible to generate cases in which conventional techniques would fail (as defined in Sect. 3.11.2), but the proposed techniques would correctly identify targets.

As discussed in Chapter III, for the purpose of this research it was necessary to define not only target models of interest, but also target models (parameter sets) in some sense "in between" two *parent* target models. These "interpolated models" will define specific likelihood functions for target parameters in between the "endpoint" or origin targets of interest, and will allow us to construct the generalized ambiguity function curves for each basic form of likelihood function (see Sect. 3.11). The SCAMP file format allows a straightforward approach to "interpolated" targets. Each parent was defined by the same number and type of shapes and surfaces, but the locations occupied by these objects in 3-dimensional space differed according to the size and shape of the respective target. For example, the two targets in Figs. 5.2 and 5.3 are defined by the same number and type of shapes and surfaces, with only locations changed. Similarly, the three targets in Figs. 5.4 through 5.6 are commonly defined in this way. Borrowing from the evolving language of computer graphics [22], then, 3-D linear interpolation "morphs" (morphological, or shape, transformations) were performed to obtain new "points" in target parameter space, or new targets in some sense "between" the two parents.

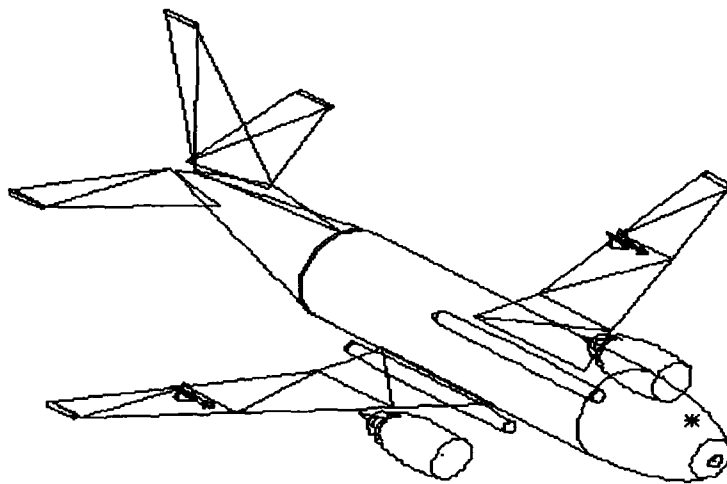


Figure 5.2. Boeing B-737

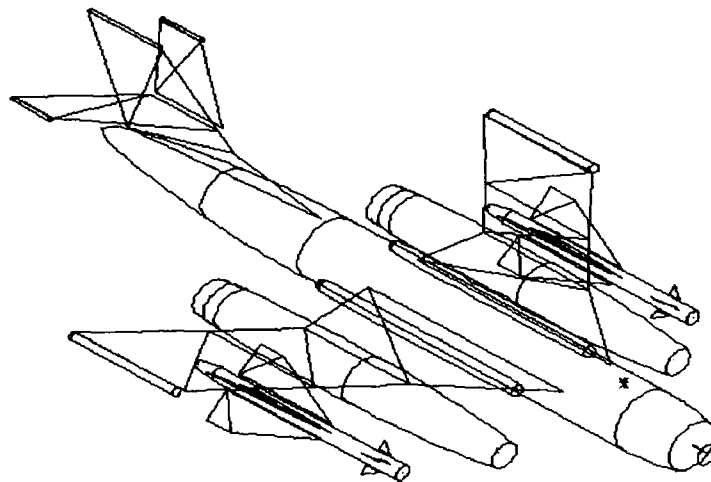


Figure 5.3. Yakovlev YAK-28

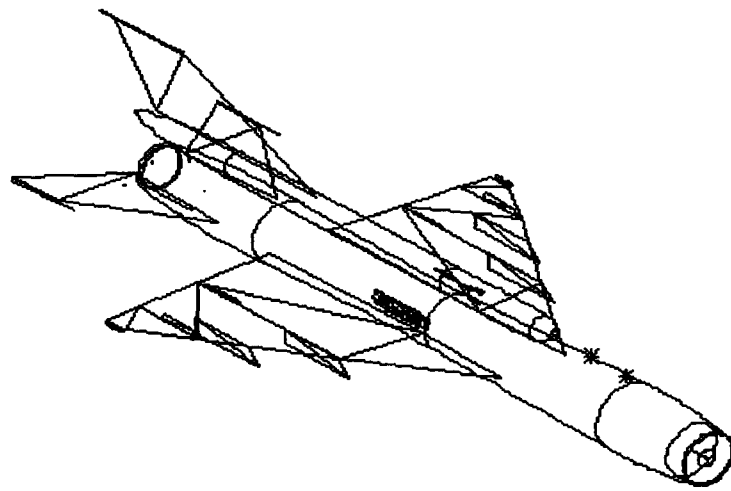


Figure 5.4. Mikoyan MIG-21

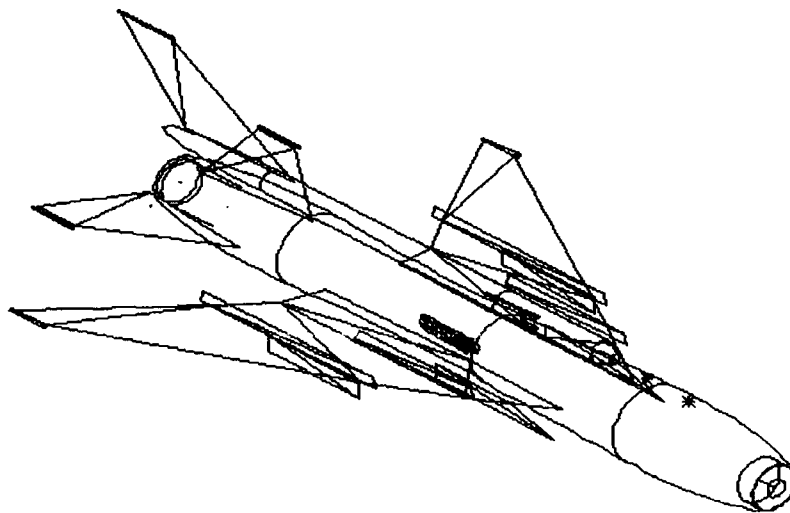


Figure 5.5. Sukhoi SU-22

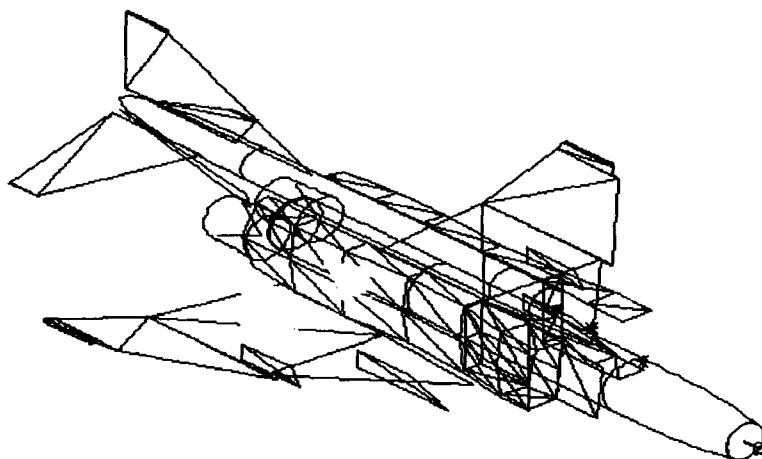


Figure 5.6. McDonnell-Douglas F-4

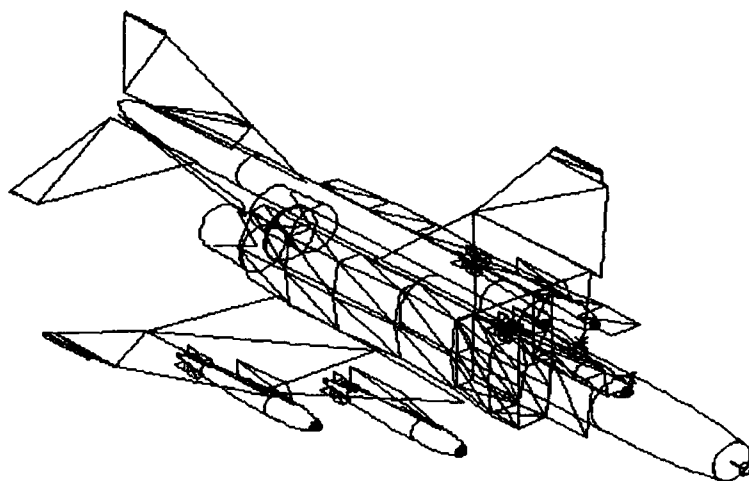


Figure 5.7. McDonnell-Douglas F-4 with stores (bombs)



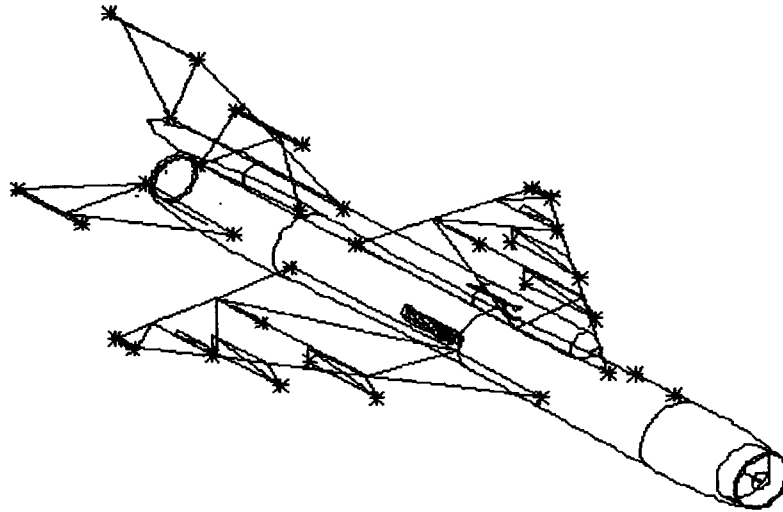


Figure 5.8. Mikoyan MIG-21 with Extra Point Scatterers

The morphing process is performed by a modified version of an original RCSToolBox subroutine that reads two SCAMP parent files line-by-line and writes output SCAMP files for desired interpolation (or extrapolation) values. The output SCAMP morphs are then post-processed somewhat in the usual SCAMP file editing process to re-smooth polyellipsoid surfaces and eliminate intersections of surfaces. Fig. 5.9 shows an F-4 Phantom II and a MIG-21 as parent targets, and a pseudo-target defined by 50% interpolation between the parents. Note the difference in size between the three models – often a key discriminant, where the target-sensor relative position can exploit it. It must be emphasized that this linear interpolation was never expected to translate into linear changes of the likelihood function outputs, and it did not.

*5.3.4 Test Data Analysis for Noise Statistics.* Like SRCRCS, RCSToolBox is deterministic. As noted in Sect. 2.2.3, however, actual signatures exhibit considerable noise, or random variation with time, even for constant aspect angle. All of the signature comparison metrics available for this research assume a random noise component in the signature. In particular, the Mahalanobis metric assumes Gaussian signature statistics,

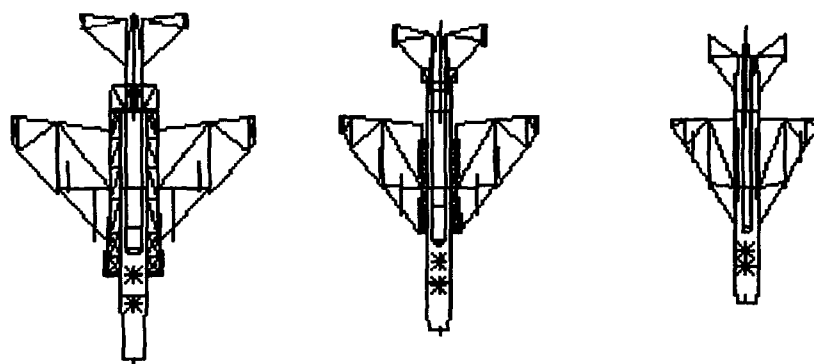


Figure 5.9. F-4, 50% Morph, and MIG-21. Note the difference in size between the three models – often a key discriminant, where the target-sensor relative position exploits size differences.

and requires a covariance matrix for computation of the weighted distance between two signatures. The Mahalanobis metric was used in all of the proposed algorithms to provide a comparison baseline, and obtaining *some* representation for HRR radar signature means and covariances was essential.

Therefore, an analysis of actual HRR radar test results was conducted using test data and software provided by the Wright Laboratory Target Recognition Technology Branch (WL/AARA) [166, 20]. The data were taken from a ground station illuminating typical aircraft in flight. Each signature encompassed hundreds of range bins over a classified range extent, and hundreds of signatures were available for areas of small angular extent or windows (say  $10 \times 10$  degrees) on any given test set. Signatures were coded as to discrete aspect angle value within each window, using aspect angle information recorded by telemetry on the target aircraft. Thus, statistical information like means and covariance matrices can be developed for signatures within the whole window or sub-windows on each data file.

Three signatures chosen at random from a typical data set of this research were shown in Fig. 2.1 in Chapter II. These results are from a  $7 \times 10$  degree window over which 484 signatures were taken. Each of these signatures is actually the result of summing returns from several dozen individual pulses taken over a period of much less than one second.

The 128 range bin amplitude values shown are found by cropping a number of bins off each end of the original signature, downsampling the remaining bins to 256 bins by taking the maximum amplitude in each set of three bins, and finally cropping bins as required at either end of the 256 bin extent to center the signature in the remaining 128-bin extent. The length of the signatures is of the order of the size of an aircraft, so that the bin widths lie in the range of 0.05 to 0.5 meters. The centering process is performed essentially by correlating each signature with the mean of all previous signatures.

These signatures are not normalized for total energy. Normalization has been used to remove signature differences due to total energy, making the fundamental shape of signatures for a single target more consistent with changes in aspect angle, or attempting to reduce uncertainties due to imprecisely-known atmospheric transmission coefficients, radar calibration, etc. However, the normalization can also increase ambiguity in comparing two different targets. At this time, it appears that some form of energy normalization may be required in HRR radar target recognition [166, 56].

The aspect angle window for Fig. 2.1 is positioned approximately in the plane of the wings looking at the rear of the aircraft. The leftmost portion of this figure corresponds to the rear of the target, and the figure extends in range to the right toward the front of the target. It appears that there are three fairly strong and consistent scatterers in the area of range bins 40 to 60. Since this is a jet engine aircraft, the HRR radar signature from this tail-end aspect strongly exhibits the effects of multiple bounces within the rear engine cavity – the “trailing” returns in the area of bins 80 to 100 are likely due to cavity effects.

Figures 5.10 through 5.12 show means and variances for collections of signatures over three different aspect angle windows and three different aircraft. Fig. 2.1 and 5.10 correspond to the same signature set. Note that the three scatterer locations evident in Fig. 2.1 stand out “in the mean” on Fig. 5.10. Fig. 5.11 shows means and variances from a set of returns over a window approximately 20 to 30 degrees off the nose of the target aircraft, and Fig. 5.12 shows corresponding results from near nose-on (all signatures are taken close to the plane of the wings). Since the origin target classes are unknown (classified secret), it is difficult to say precisely how the major peak locations correspond to physical structures on the targets. In Fig. 5.12, however, we can be reasonably certain

that the first peak on the left corresponds to the forward fairing or bulkhead of the aircraft frame. Remember that the forward fairing is of necessity transparent to radar waves if the target aircraft is equipped with a nose-mounted radar (in which case the fairing is a *radome* – note that this fact was considered in the earlier modelling efforts, e.g., Fig. 5.7). The next peak to the right, then, may correspond to scatterers from the cockpit area, and so on.

It is important to restate and emphasize that the downsampling technique employed here was to pick the *maximum* amplitude value from the three values corresponding to three contiguous original bins. Other researchers have simply discarded two out of three bin amplitudes – possibly discarding a high amplitude return in favor of a null value in a neighboring bin. It is the belief of the author and others [19] that “pick maximum amplitude from  $n$  bins” downsampling (where  $n = 3$  in this case) is more likely to trap the desired information. If a “pick every  $n$ th amplitude” approach is used instead, the mean will be lower, and the variance higher, just as we would expect.

In any case, these plots show clearly that, even though individual signatures are highly noisy, there is a definite statistical consistency in the signatures, even over these relatively large aspect angle windows. They also support the conclusion that variance can be treated as reasonably constant across the target range extent for a variety of targets and aspect angles. Clearly, these results do *not* support the conclusion by some [56] that HRR radar signatures “completely decorrelate” with small changes in aspect angle.

Most researchers treat HRR radar signatures as statistically independent from bin-to-bin. In this analysis, the bin variance values were actually gathered as part of the full covariance matrix for each set of HRR radar signatures, and correlation coefficients were defined to show inter-bin statistical relationships. For the 128-bin vectors shown here, correlation coefficients were on the order of 0.7 for adjacent bins, dropping to roughly 0.3 for separations of five bins, and to roughly 0.2 for separations of ten bins. Thus, there is significant bin-to-bin correlation, but this is to be expected due to (1) physical relationships between the scatterers in adjacent range bins, (2) processing in the radar itself, and (3) errors in signature alignment or range registration. Certainly, some portion of the variance itself in these signatures arises from the errors in range registration.

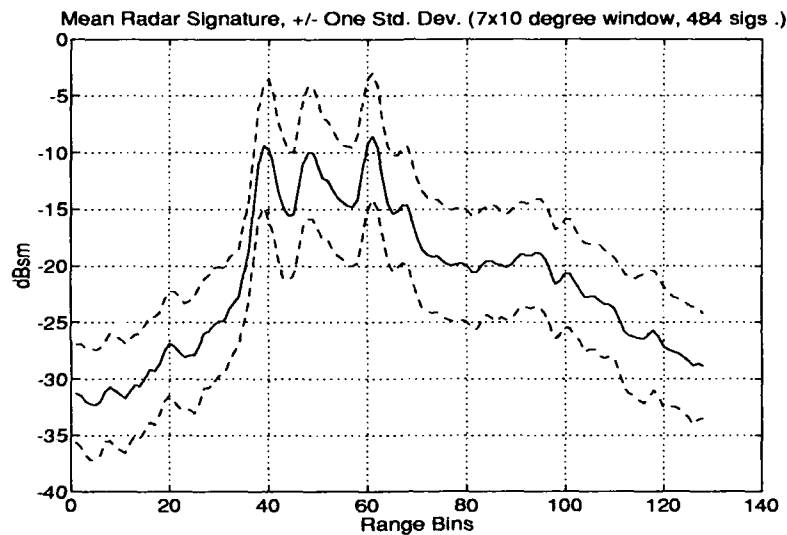


Figure 5.10. Measured Signatures: Means and Variances – Rear Aspect (test data taken from tape GTw11Atran.dathr, results of [20], provided by [166], with further processing by the author)

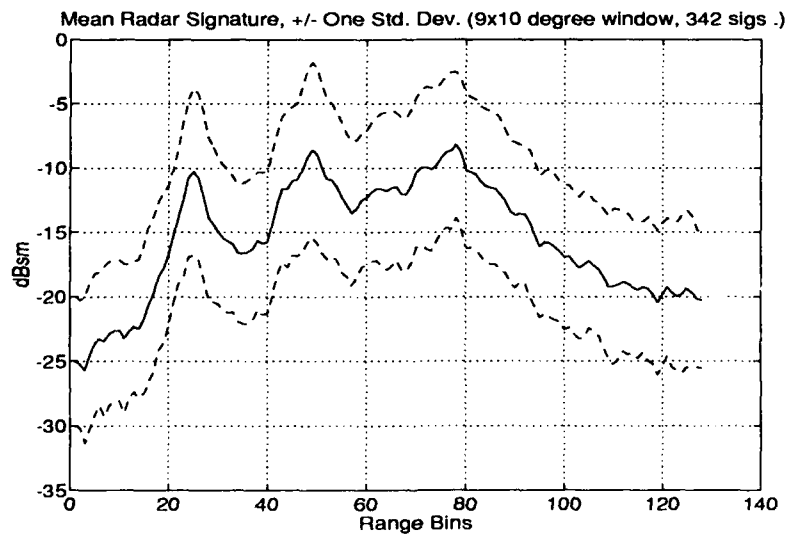


Figure 5.11. Measured Signatures: Means and Variances – Front/Side Aspect (test data taken from tape GTw39Atran.dathr, results of [20], provided by [166], with further processing by the author)

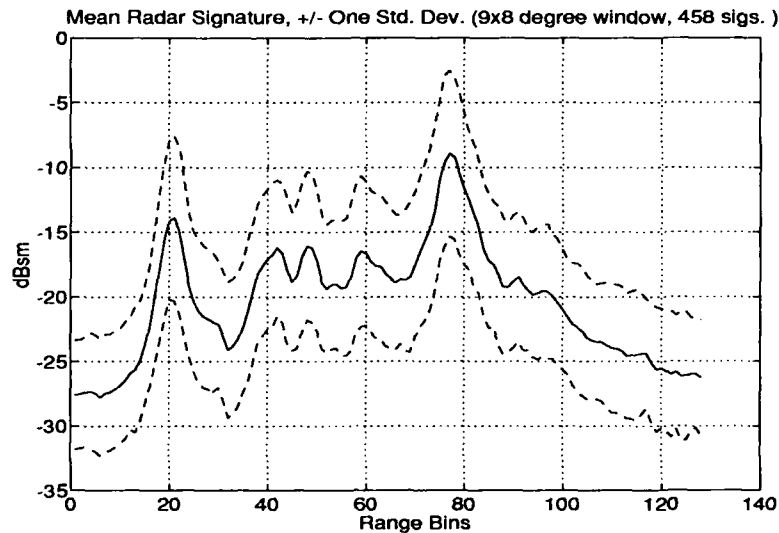


Figure 5.12. Measured Signatures: Means and Variances – Front Aspect (test data taken from tape GTw49Atran.dathr, results of [20], provided by [166], with further processing by the author)

Where the individual bin amplitudes are treated as Gaussian and a Mahalanobis (log Gaussian likelihood) metric is employed, however, the assumption of bin-to-bin independence allows the Mahalanobis distance computation to be reduced from a matrix-vector and vector-vector (inner product) multiplication to two inner product multiplications. Assumption of constant variance across the target extent allows further reduction to one inner product multiplication and a scalar division. Since the range registration process in general will require these operations to be performed several times to find a maximum value for any one association of an observed signature with a library signature, these opportunities for reduction in on-line computation are important. In particular, as noted in Sect. 2.2.3, the assumptions of bin-to-bin independence and constant variance allow this comparison to be done only once in the frequency domain (this research was performed using range domain alignment).

It is clear, however, that researchers generating noise samples for addition to some mean to provide a truth signature or realization should not lightly assume bin-to-bin inde-

pendence. In this research, the bin-to-bin independence assumption was made consciously to reduce computation in the simulation process.

Other investigation not shown in detail here supported the practice of treating the noise in any one bin as Gaussian in units of decibels (to the extent that errors in range registration allow us to examine processes in a *single* range bin), but this does not imply that the HRR radar signature vector can be treated as a *multivariate* Gaussian process. Treating radar cross section as Gaussian in decibels is equivalent to the use of the classical "lognormal" statistical model for cross section in units of area [15:114], but the author has found no formal analysis to the effect that this is a valid model for HRR radar cross section. In some sources (e.g., [92]), no clear indication was given as to whether or not radar cross section data treated as Gaussian was in fact given in units of square area (i.e., proportional to signal power) or log square area. Since it was desired to model signal statistics at least to first order, this small effort was expended to gain confidence in the use of dBsm as a unit of measure – the intent, however, was only to obtain some probabilistic signature description, in whatever units.

Other issues of statistical interest not addressed here are the degree of independence of signatures over time, with or without changing aspect angle, and the possibility of multiplicative error sources. Our research will assume that the signatures are a function only of target class, incident radar waveform, aspect and polarization angles, and that noise in each HRR radar signature bin is additive, independent from bin-to-bin, white (independent in time) and Gaussian in dBsm. Noise standard deviations from five to nine dBsm were used. As shown in Figs. 5.10 through 5.12, standard deviations of five to six dBsm are supported by experiment: higher levels were used to increase ambiguity, thereby accentuating improvements from the proposed recognition methods.

In conclusion, note that detailed statistical analysis of HRR radar signatures was *not* an objective of this research – this subject was pursued only far enough to provide reasonable information for simulation studies. The author believes, however, that additional research in this area is much needed.

*5.3.5 Generating Simulated Signatures.* The results of the previous section have provided a general understanding of observed statistics for actual HRR radar tests on typical aircraft. This research used these statistics in two ways: (1) to generate noise samples for addition to the deterministic bin amplitudes given by RCSTooLLBox, and (2) to define (diagonal) covariance matrices for use in Mahalanobis metrics. The RCSTooLLBox signatures defined the "mean" signatures, and the test data-defined noise covariances provided baseline statistics for noise generation and definition of the Mahalanobis (covariance) weighting matrix.

Each of the following figures contains two curves. Figures 5.13 through 5.15 contain: (1) the deterministic RCSTooLLBox signature for a given target model (the F-4 shown in Fig. 5.6), aspect angle, polarization angle, center frequency (10 GHz), and bandwidth (1 GHz), and (2) one noise-corrupted signature, given by the sum of the deterministic signature with a vector of independent noise realizations generated consistently with the statistics defined above (a standard deviation of 5 dBsm, constant for all bins). The last of the signatures generated under these assumptions, Fig. 5.16, is the counterpart of Fig. 5.14 for the MIG-21 model shown in Fig. 5.4; that is, a signature that would be generated for a MIG-21 flying the same trajectory (accounting for fundamental differences in the aerodynamics of the MIG-21 and F-4). Radar center frequency and bandwidth figures were selected to be consistent with parameters observed to be in use by radar researchers under contract to Wright Laboratory [21].

These signatures are taken generally from a nose-on aspect angle: the forward part of the target corresponds to the left part of the signature, and the rear part of the target to the right part of the signature. Each range bin is approximately 0.225 meters long. In particular, the two peaks at -10 dBsm just after bin number 40 on the F-4 models are due to cockpit scatterers, and the rather large return at range bin 80 is due to the rear engine structure. In the MIG-21 signature, only one cockpit scatterer is present, and the engine structure has largely disappeared, leaving only a small return from the tail.

The reader will observe that signatures Figs. 5.14 and 5.16 exhibit some similarities, but remain distinctly different – particularly in overall length of the signature, owing to the large difference in length between the F-4 (18 meters) and the MIG-21 (13.5 meters). The



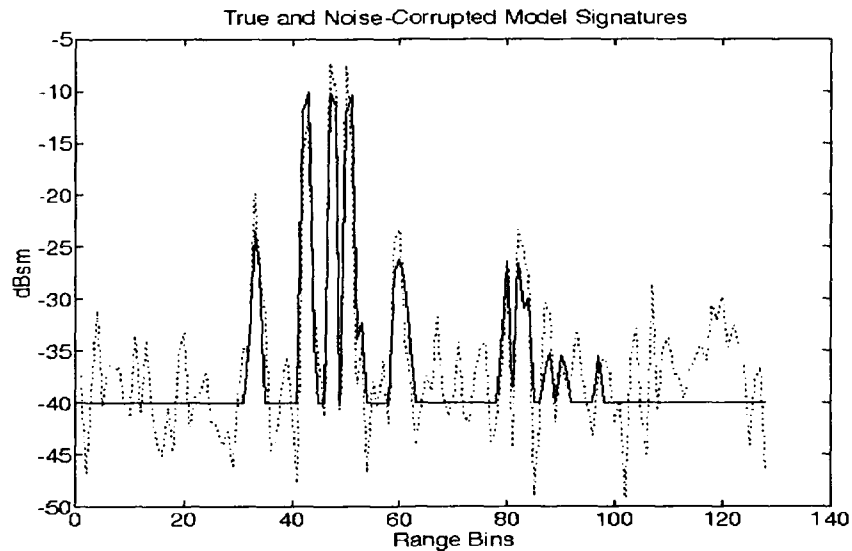


Figure 5.13. Simulated Signatures: RCSTooLLBox-Generated (Solid Line) and Noise-Augmented (Dotted Line). F-4 Model, Frontal Aspect, 10 GHz Center Freq., 1 GHz Bandwidth, 5 dBsm Std. Dev. Noise.

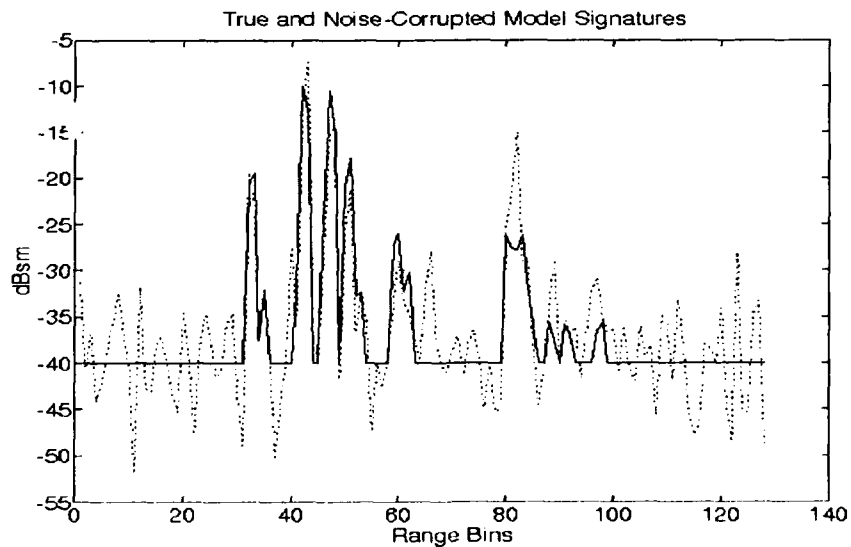


Figure 5.14. Simulated Signatures: RCSTooLLBox-Generated (Solid Line) and Noise-Augmented (Dotted Line). F-4 Model, Frontal Aspect, 10 GHz Center Freq., 1 GHz Bandwidth, 5 dBsm Std. Dev. Noise (displaced from signature in Fig. 5.13 by 0.8 sec., 1.83 deg.).

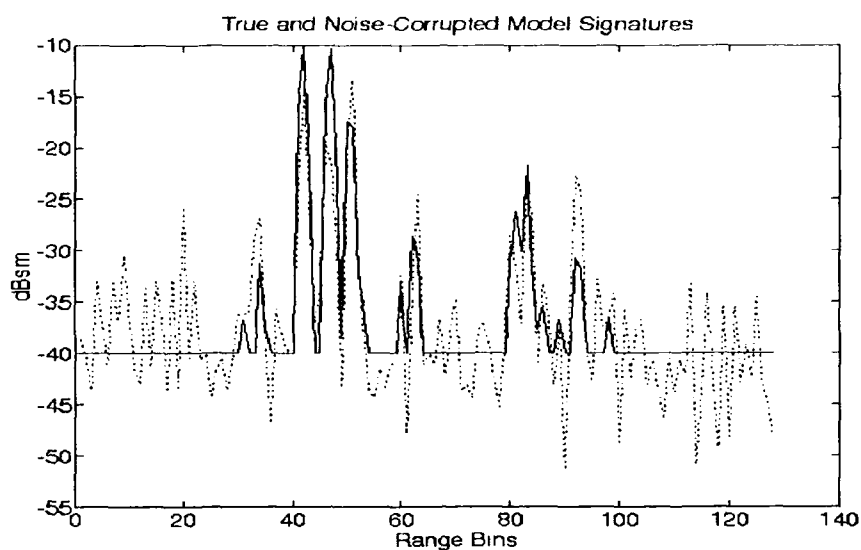


Figure 5.15. Simulated Signatures: RCSToolBox-Generated (Solid Line) and Noise-Augmented (Dotted Line). F-4 Model, Frontal Aspect, 10 GHz Center Freq., 1 GHz Bandwidth, 5 dBsm Std. Dev. Noise. (displaced from signature in Fig. 5.14 by 0.8 sec., 1.83 deg.).

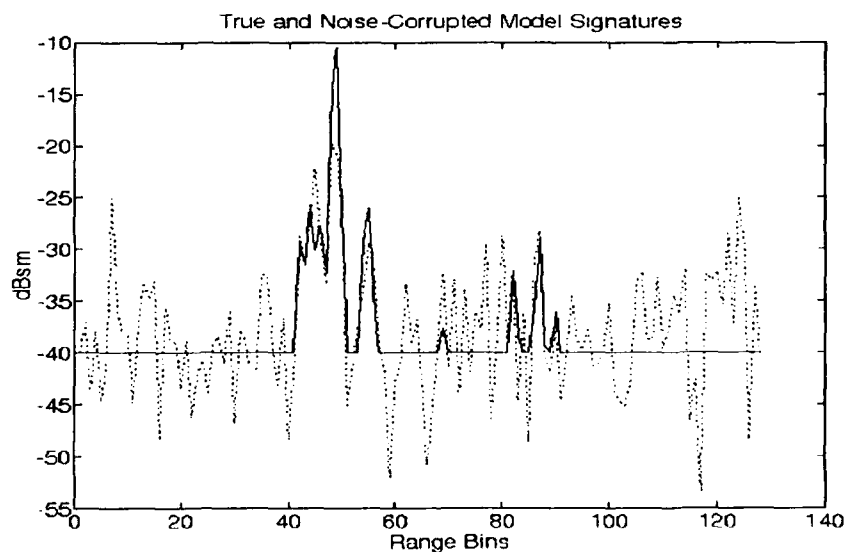


Figure 5.16. Simulated Signatures: RCSToolBox-Generated (Solid Line) and Noise-Augmented (Dotted Line). MIG-21 Model, Frontal Aspect, 10 GHz Center Freq., 1 GHz Bandwidth, 5 dBsm Std. Dev. Noise (corresponds for same kinematics to F-4 signature in Fig. 5.14).

Mahalanobis metric and other HRR radar signature metrics are very sensitive to signature length, and, even with large standard deviation noise added, conventional (independent look) algorithms operating on these *simulated* signatures have little trouble selecting the correct target class. For *real* signatures, "multibounce" effects from the engine intake cavities and similar effects would make these length differences less noticeable, and this would increase ambiguity considerably.

Ultimately, sufficiently challenging recognition tests were found by defining scenarios satisfying these conditions: (1) target models augmented with extra scatterers; (2) models scaled to similar size and/or viewed at aspect angles where target size was not a factor; (3) reduced radar bandwidth, so that range resolution was reduced; and (4) Gaussian signature noise of standard deviation eight or higher (i.e., rather greater standard deviation than values observed in actual data). The decision to reduce radar bandwidth was made after consultation with Mr. Walter Barnes [11] of WL/AAR, who indicated that the 1 GHz bandwidth figures common to recent research efforts might not be practical in some implementations, but that a reasonable simulation baseline would be approximately one-third of the 1 GHz figure.

Figures 5.17 through 5.19 are typical of these "challenging" signatures. These are taken from an aspect angle in which the plane of the wings is nearly normal to the sensor-to-target or boresight vector, so that the overall range extent of the signature is minimized. Figs. 5.17 and 5.18 are taken from the MIG-21 target model in Fig. 5.8, while Fig. 5.19 is taken from a version of the SU-22 as in Fig. 5.5, but downscaled 20% to the size of the MIG, and equipped with scatterers in locations corresponding to those of the MIG, with the same scatterer radar cross sections. The aspect angle of Fig. 5.19 corresponds exactly to that of Fig. 5.17 (identical aerodynamic characteristics are assumed). Finally, consistent with the comments in the previous paragraph, the radar bandwidth for these signatures was 333 MHz (10 GHz center frequency, as above) and noise standard deviations are set to nine dBsm.

Comparing Figs. 2.1 and 5.10 through 5.12 to Figs. 5.13 through 5.19, the reader will observe that the apparent "noise floors", or mean amplitudes for range bins that evidently correspond to free space (atmosphere), are generally lower for the simulated signatures than

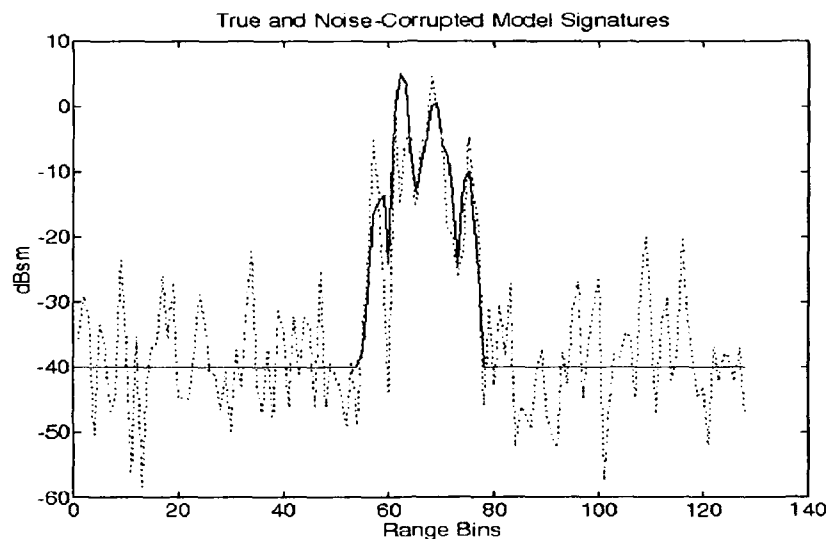


Figure 5.17. Simulated Signatures: RCSToolBox-Generated (Solid Line) and Noise-Augmented (Dotted Line). MIG-21 Model, Extra Scatterers, Top Aspect, 10 GHz Center Freq., 333 MHz Bandwidth, 9 dBsm Std. Dev. Noise.

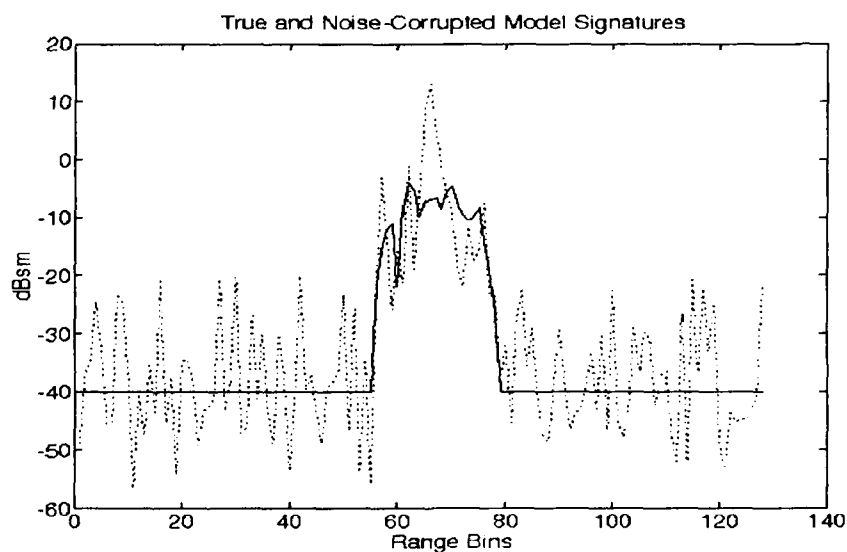


Figure 5.18. Simulated Signatures: RCSToolBox-Generated (Solid Line) and Noise-Augmented (Dotted Line). MIG-21 Model, Extra Scatterers, Top Aspect, 10 GHz Center Freq., 333 MHz Bandwidth, 9 dBsm Std. Dev. Noise. (displaced from signature in Fig. 5.17 by 0.8 sec., 3.67 deg.).

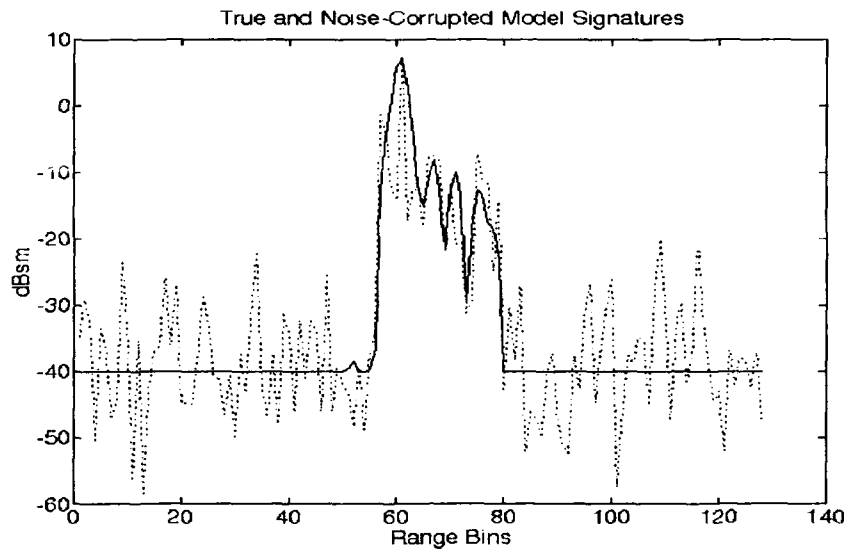


Figure 5.19. Simulated Signatures: RCSToolBox-Generated (Solid Line) and Noise-Augmented (Dotted Line). SU-22 Model, Extra Scatterers, Top Aspect, 10 GHz Center Freq., 333 MHz Bandwidth, 9 dBsm Std. Dev. Noise. Corresponds for same kinematics to MIG-21 signature in Fig. 5.17

for the observed signatures. This is by choice. The deterministic noise floors returned by RCSToolBox for the models shown above tend to lie between -50 and -80 dBsm. The author consciously raised the noise floor for simulated outputs to values from -30 to -45 dBsm to provide a signal structure, or minimum-to-maximum amplitude range comparable to that of the measurements. Since the peak simulation returns in general tend to be lower than those for available test data (presumably due to the specular model effects noted in Sect. 5.3.2), raising the simulation noise floor to the test-observed values of -30 to -35 dBsm would have left little signal structure for comparison.

For a few aspect angles, however, the specular model associations will produce very high amplitude returns. It should be recalled from Sect. 2.2.3 that the range sweep generating process can be viewed as the convolution of a sinc-shaped radar pulse in range/time (i.e., a rectangle in frequency with chosen center frequency and bandwidth) with an array of scatterers represented as impulse (i.e., Dirac delta) functions. At certain aspect angles, a scatterer may possess a particularly high cross-section, or impulse function magnitude. For example, consider a flat, perfectly conducting plate normal to the incidence direction

from a monostatic radar (i.e., a radar in which the transmit and receiver antennas are co-located). The convolution process, then, will result in "range sidelobes", or "false" peaks in the HRR radar signature, due to convolution of the high cross section scatterer with side lobes of the sinc-shaped pulse.

This effect is observed in simulations and tests, but the specular nature of RCSToolBox models mean that the effect can be so pronounced for some models and aspects (e.g., the F-4 model in Fig. 5.6 at side-on aspects) that the HRR radar signature is raised to +30 dBsm or more in *every* range bin. Since the signature so generated looks like a wall, the author refers to this effect colloquially as the "wall" effect. Due to effects of this nature, the author inserted code to identify and skip "wall effect" signatures, and to limit or "clip" individual high amplitude peaks. Clipping high amplitude peaks also tends to increase ambiguity, which helps to demonstrate the proposed methods, as well as to provide more realistic signatures.

**5.3.6 Summary.** This section has described the methodology by which simulated signatures were generated for this research. The process selected, after consideration of trade-offs for model availability, time, and fidelity, was to generate signatures with RCSToolBox using models that resemble tactical targets of interest to first order. Measurement realizations were generated by adding zero-mean, independent (from bin-to-bin), white (independent in time, or from signature to signature) Gaussian noise to the RCSToolBox "mean" signatures for true aspect angles on the model chosen to be the "true" target. Noise statistics were fundamentally consistent with those for observed signatures.

#### **5.4 Initial Concept Demonstration**

**5.4.1 Test Objectives.** As a preliminary confidence-building test of the basic motion-warping concept, an early demonstration was conducted using several key tools planned for use in the proposed research. The objective of this research was to investigate the potential for applying dynamic programming sequence comparison techniques to sequences of high range resolution radar signatures. Significant questions were:

- (1) Would a classical sequence comparison or dynamic time warping algorithm work at all with a high range resolution radar signature comparison metric and sequences of range sweeps?
- (2) Would similar (close in an aspect angular sense) but non-identical sequences from the same target produce small warping path distances?
- (3) Would matches over the same or similar aspect paths between significantly different target models produce significantly larger matching path distances than those seen in the preceding case (2)?

The tools used in the test were:

- (1) The Syracuse Research Corporation (SRC) HRR radar range sweep generator "SRCRCS" [53] - RCSToolLLBox was not yet available at the time of this initial set of tests.
- (2) Aircraft target models for use with SRCRCS, provided by Wright Laboratory (WL/-AARA) [166].
- (3) The General Dynamics "slide distance" metric for inter-sweep "distance" [94], as discussed in Sect. 2.2.3.
- (4) A simple time warping algorithm given in App. B of [176], with modification by the author to reduce the cost penalty for warping path length. Specifically, we multiply the added (new association point) cost for vertical and horizontal transitions (see Fig. 2.8) by a factor of  $\frac{1}{2}$ , so that the sum of added costs for one vertical and one horizontal transition adds roughly the same cost as one diagonal transition, as discussed in [195].

This test did not require or employ a target tracker or aspect angle estimator algorithm, since such algorithms had been successfully tested by others in the past [5, 77, 120, 121].

*5.4.2 Test Procedures.* The test was begun by generating sequences of range sweeps in the SRCRCS model using the "RTI" (Range-Time Intensity) option [53]. This option provides radar range sweeps over a user-specified angular extent (some portion of a "great circle" on the hypothetical aspect angle sphere) at discrete angular values. Pairs of such sequences were then processed to extract peaks and the processed sequences

were warped against each other, using the slide distance metric as an inter-sweep distance measure.

Figs. 5.20 through 5.22 show three typical range sweep sequences, covering aspect angular extents of 180, 90, and 90 degrees respectively over different paths, for aircraft models similar to the F-4 in Fig. 5.6. Note that each plot consists of 37 "sweeps" at discrete angle values, listed by sweep number along the lower right axis. Each sweep consists of radar cross section values in 128 range bins, shown along the bottom axis. The vertical axis shows radar cross section in decibel square meters (dBsm). The sensor can be thought of as lying to the extreme left, so that the leftmost returns are closest to the sensor.

With reference to the SRC X-Y-Z target coordinate frame in Fig. 5.1, Figs. 5.20 through 5.22 represent respectively the following basic aspect angle paths, typical of those used in the test:

- (1) Fig. 5.20: a 180 degree angular extent in the SRC X-Y plane, centered on the SRC Y axis, with 37 sweeps taken at 5.0 degree increments. Thus, the angular extent of Fig. 5.20 lies in the plane of the wings, moving in azimuth angle from the right to left side of the target. The center signature, or number 19 in the sequence from 1 to 37 as marked in the figure, is nose on to the target. These signatures are taken from a generic F-4 model.
- (2) Fig. 5.21: a 90 degree angular extent in the plane of the wings, centered on the SRC X axis, with 37 sweeps taken at 2.5 degree increments. Therefore, this angular extent lies from the right front (45 degrees azimuth) to right rear (135 degrees azimuth) of the target. The center signature, or number 19 in the figure, is side on to the target. These signatures are taken from a generic F-14 model.
- (3) Fig. 5.22: a 90 degree angular extent in the SRC X-Z plane, centered on the SRC X axis, with 37 sweeps taken at 2.5 degree increments. Therefore, the 90 degree extent covered by Fig. 5.22 lies in the plane which bisects and is normal to the longitudinal axis of the target, moving from -45 to +45 degrees in elevation. The center signature, or number 19 in the figure, is side on to the target. These signatures are taken from a generic F-16 model.



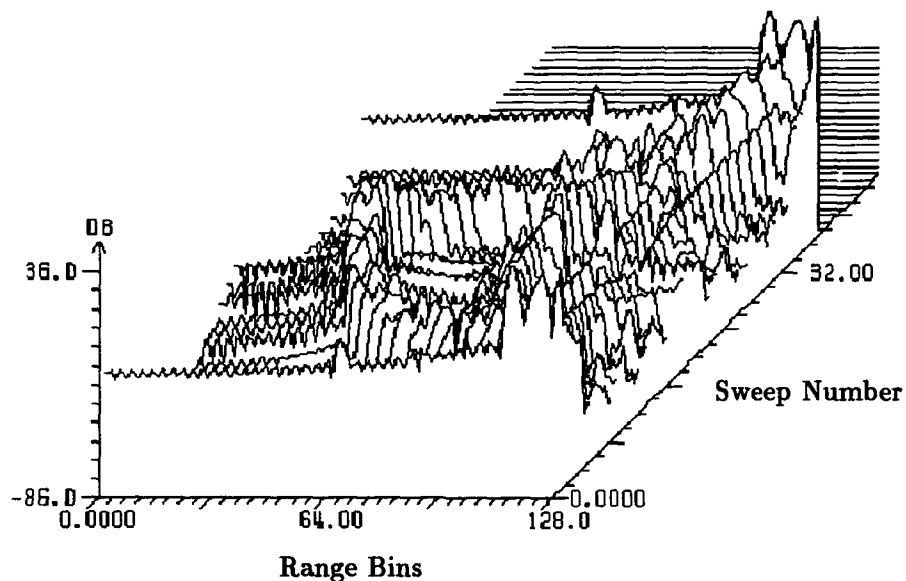


Figure 5.20. Typical Range Sweep Sequence – F-4 model, 180 degree angular extent in the plane of the wings, moving from the right to left side of the target, centered on the target nose, with 37 sweeps taken at 5.0 degree increments.

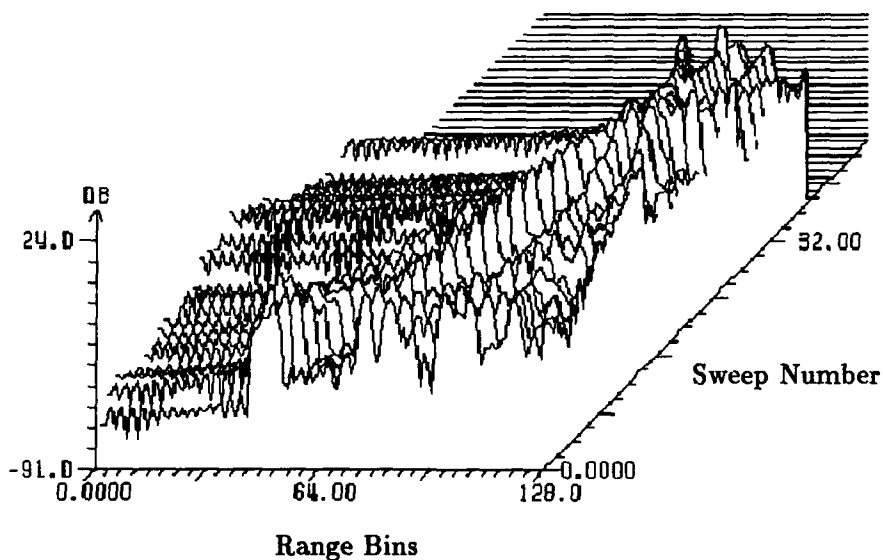


Figure 5.21. Typical Range Sweep Sequence – F-14 model, 90 degree angular extent in the plane of the wings, moving from the right front to right rear of the target, centered on the right wing, with 37 sweeps taken at 2.5 degree increments.

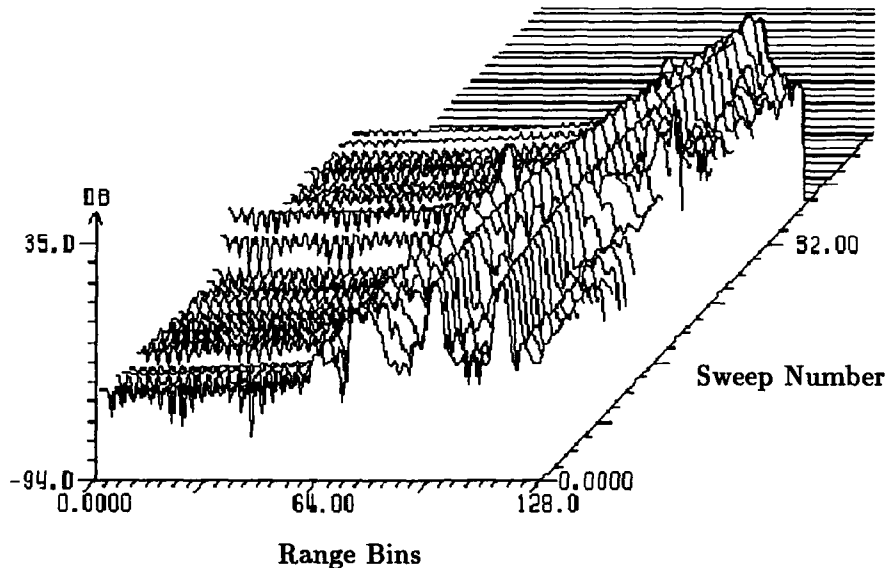


Figure 5.22. Typical Range Sweep Sequence – F-16 model, 90 degree angular extent normal to the plane of the wings, moving from the lower right to upper right of the target, centered on the right wing, with 37 sweeps taken at 2.5 degree increments.

These signatures were produced with a center frequency and bandwidth of 1.25 GHz and 1.0 GHz, respectively (taken from the example in the SRCRCS User's Manual [53:18]). Note that Fig. 5.20 exhibits the “wall effect” discussed in Sect. 5.3.5 – these high-return signatures at sweeps 1 and 37 correspond to the left and right sides of an F-4 model in the plane of the wings, where the radar cross section is quite large.

Readers familiar with speech processing will observe the similarity between these three figures and frequency vs. time plots of human speech – viewed from above, the migrating peak locations (heuristically, “mountain ranges”) in these plots bear an uncanny resemblance to *formant tracks* [176:121–123], or high energy bands over time, in frequency vs. time plots of speech. Formants are frequently used as features for dynamic time warping in speech recognition [176:297], and the existence of similar structures in these radar signatures (bearing in mind that true signatures have much more noise) was a strong inducement to continue research.

Three basic types of tests were conducted: (1) sequences extracted from different target models over the same aspect angle range were warped against each other; (2) sequences

extracted from the same model, but over slightly different paths, were warped together; and, (3) sequences from different targets over slightly different aspect angle paths were warped together. Additionally, some clearly undesirable matches were attempted as well, matching two sequences taken from completely different aspect angle paths over the same or different models. The tests did not consider noise.

*5.4.3 Test Results.* All in all, the test was very successful. The answer to each of the questions asked above was *yes*.

(1) Dynamic "time" warping or classical sequence comparison did work with the slide distance metric and sequences of range sweeps. The sequence expansions and compressions characteristic of warping processes were observed where expected in these tests.

(2) Similar (close in an aspect angular sense) but non-identical sequences from the same target did produce small warping path distances. Warping path distances between 0 (zero) and 60 slide distance units (SDU) were generally observed for such cases.

(3) Matches over the same or similar aspect paths between significantly different target models did produce notably larger matching path distances than those seen in the preceding case (2). Warping path distances in excess of 150 SDU were generally observed for such cases.

Some potential but expected problem areas were identified. In particular, it was noted that for similar target classes (e.g., an F-14 and an F-15), the warping algorithm may provide a closer warping distance for mismatched aircraft over a given identical aspect angle path than for the same aircraft over slightly different (say 10 degrees "off-nominal path") aspect angle paths. This was an early indication that "off-nominal path" errors could be significant, and that a single "one-dimensional" sequence comparison for any given target would not be adequate.

This observation can be stated in terms of expected ambiguity function behavior (see Sect. 3.11.3) for movements or changes in aspect angle *state* and target *parameter* spaces. It appears, for example, that the generalized ambiguity function for (1) comparison of signatures from one path relative to those from other "close" paths on the *same* aircraft (i.e., changes in aspect angle state history) may be more sharply peaked (i.e., exhibiting

more downward curvature) than the generalized ambiguity function for (2) comparison of this signature sequence to those from the same path over other (similar but *different*) aircraft (i.e., changes in shape parameters). Clearly, this observation is valid only for the observables, distance metric and aspect angle/aircraft mismatch ranges examined here.

Another problem was that the conventional warping algorithm used in this test allowed no warp flexibility at the endpoints of the sequences, which were constrained to match. For one case in which an F-4 model-derived sequence was matched against a slightly offset sequence, a very high matching cost was obtained. This was almost certainly due to the fact that the endpoints of one sequence were the high amplitude returns from the sides of the F-4, while the endpoints of the other sequence were radically different, and high costs from this endpoint mismatch could not be avoided. Ultimately, then, an "unrestricted endpoint" technique [182], as applied in speech recognition for words with uncertain start- and endpoints, was applied for classical sequence comparison-type techniques to overcome this problem. Larson and Peschon-type techniques, not forced to match each element along an aspect angle path, inherently have "unrestricted endpoint" qualities.

**5.4.4 Conclusion.** In sum, this short test gave early confidence that dynamic programming sequence comparison techniques could be applied to high range resolution radar signatures.

## **5.5 Detailed Procedures for Motion Warping**

**5.5.1 Introduction.** This section presents a detailed discussion of procedures used in this research to apply dynamic programming sequence comparison techniques in target recognition. These algorithms are basic applications of "motion warping" as defined and discussed in Chapters I and III – elements in the second of three classes of kinematic-feature observable fusion algorithms for object recognition, as proposed in this research. In this section, we will develop procedures more specifically, referring to previous sections where needed. Results are presented in the following Sect. 5.6.

**5.5.2 Tracking the Target.** The first step in "motion warping" is to estimate the target translational and rotational motion, or 6-DOF kinematic state. In this scenario for aircraft targets, a conventional extended Kalman filter (see Sect. 2.3) and associated fixed lag smoother (see Sect. 2.3.1.2) will estimate target position, velocity, and acceleration. The filter (tracker) development for this chapter uses conventional "kinematic" measurements of target position (range and pointing angle) and range rate (doppler velocity) only – we do not assume availability of pose estimates or other information regarding target acceleration state or aspect angle.

To estimate rotational states or (equivalently here) aspect angle from translational kinematics for aircraft targets, it is essential to determine the magnitude and direction of the target's normal load acceleration (total target acceleration normal to the velocity vector, less gravity) over the time of interest. Use of the smoother is therefore critical, due to our inability to observe the pilot's commanded attitude changes in near-real time, as Kendrick *et al.* could using pose estimates (see Sect. 2.3.3). Smoother equations (and results) are given in App. C.6.

The extended Kalman filter employed here has a standard nine-state filter model – target position, velocity, and acceleration along each of three orthogonal, assumed inertial axes in the filter frame. The target acceleration model is the standard Singer model as discussed in Sect. 2.3.2.1. Following the conventions for this model, the target acceleration standard deviation along each axis is assumed to be 32 feet per second squared, and the target maneuver correlation time is assumed to be 3 seconds. Consistent with the discussion in Sect. 2.3.2.1, these parameters define the filter dynamics driving noise strength  $\mathbf{Q}$  for the acceleration states – no pseudonoise was added to other states.

The measurement model assumptions for target position and range rate are consistent with those presented for the filter model in the previous chapter. Basically, (1) measurements are assumed to be available every 0.1 seconds, (2) target position error is modeled as white, Gaussian noise of standard deviation 100 feet, and (3) range rate error is modeled as white, Gaussian noise of standard deviation 100 feet per second. Ownship position and the direction and magnitude of gravity are assumed to be known perfectly –

the amount of error in target aspect angle estimates from these assumptions is negligible compared to error contributions from target position and velocity measurement noise.

Ultimately, it was found that even acceleration estimates from the smoother were too "noisy" to provide the desired accuracy and smoothness in estimates of target acceleration. The smoother provided an excellent estimate of target position, however – much improved over the raw filter position estimate, and robust to mismatches in target model parameters. For that reason, target acceleration was estimated by fitting target position in each inertial dimension to a second degree polynomial curve, and differentiating the curve parameters twice to derive acceleration.

*5.5.3 Developing the Kinematic Aspect Angle Path and Error Bounds.* The kinematic aspect angle path is important for two reasons. First, for the true target, or truth model, this path defines the aspect angle locations that generate the observed or measured signatures. Second, for candidate target models in a target recognition algorithm, this path defines a nominal set of points around which associations will be made between measured and model signatures, out to some aspect angle limit defined by an appropriate set of bounds. A typical path and bounds were shown in Fig. 2.9.a.

Both true and kinematically-estimated aspect angle paths are defined by tracing the path over time of the target-to-sensor vector, denoted  $\mathbf{r}_{t-s}^b$ , coordinatized in body frame coordinates (as shown by the superscript  $b$ ), with time derivatives observed with respect to the target body frame or hypothetical aspect angle sphere (for which derivatives a second superscript will be added to  $\mathbf{r}_{t-s}^b$ , when required). For the truth or target model, this path can be obtained directly – for the motion warping algorithm, it must be estimated based on target kinematics and known dynamic restrictions of candidate target classes.

*5.5.3.1 Developing the Kinematic Aspect Angle Path – Aircraft Targets.* The effort described in this section has two basic objectives. First, we seek to define a nominal or kinematically-estimated aspect angle path and path angular rate, for a given target model using some set of kinematic measurements. Second, using information developed for the first objective, the latter part of this section will establish estimated covariances for

the nominal angles and their rates. These covariances will be used to define aspect angle windows, motion warping path constraints, and aspect angle state transition likelihoods.

See Figs. 5.23 and 5.24 for vector and angle definitions in the following discussion. Recall that the velocity frame  $\vec{x}_v$  vector is defined by the direction of the flight path, that the  $\vec{y}_v$  vector is defined to lie normal to  $\vec{x}_v$  in a plane parallel to the local horizontal (the  $\vec{n} - \vec{e}$  plane), and that  $\vec{z}_v$  is defined pointing generally downward to form a right-handed set. Note also that the perspective in Fig. 5.24 is somewhat misleading for a conventional aircraft, due to the author's artistic limitations – actually, showing a positive lift in the velocity frame  $\vec{y}_v - \vec{z}_v$  plane, as the drawing implies, should tend to show more of the ventral or bottom surface of the aircraft, due to the required angle of attack  $\alpha$ .

Using the acceleration estimate derived from the polynomial curve fit to the smoothed trajectory, the baseline (coordinated turn) system will estimate aspect angle for each potential target class, using essentially the approach prescribed for the Kendrick estimator. That is, acceleration normal to the velocity vector is assumed due to wing-generated lift and gravity only, with no component from thrust, aerodynamic side-forces, or other sources. The relationship between lift and angle of attack is given by:

$$L = \frac{1}{2} \rho_{\text{atm}} V^2 S C_{L\alpha} \alpha \quad (5.1)$$

where:

$L$  = lift force magnitude (aircraft mass assumed known)

$\rho_{\text{atm}}$  = atmospheric density

$V^2$  = velocity magnitude squared

$S$  = aerodynamic surface area

$C_{L\alpha}$  = coefficient of lift

$\alpha$  = angle of attack

Thus, for these conventionally-controlled aircraft classes, normal load magnitude will determine angle of attack for a given aircraft class and velocity, and normal load direction

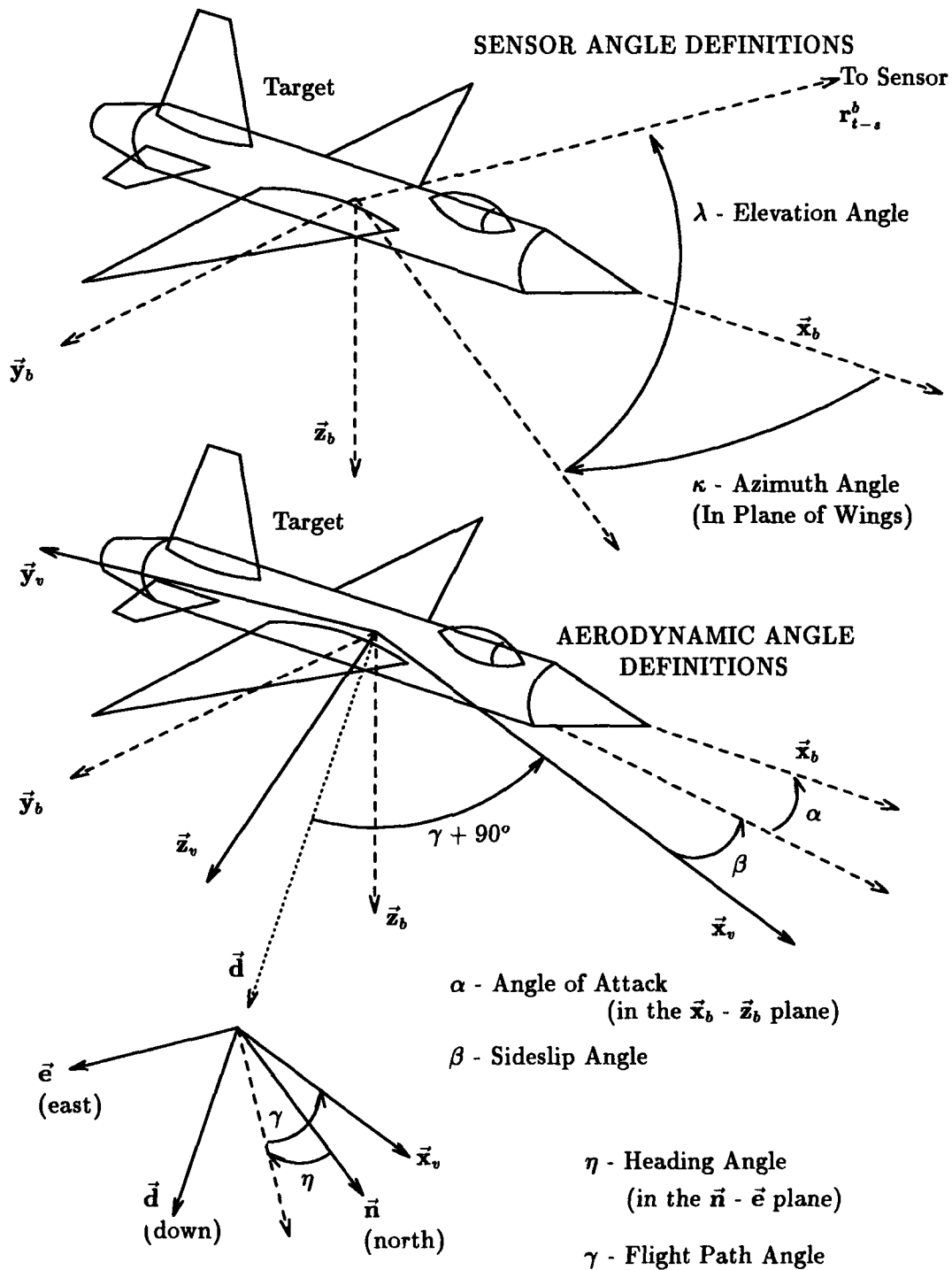


Figure 5.23. Coordinate Frame Definitions



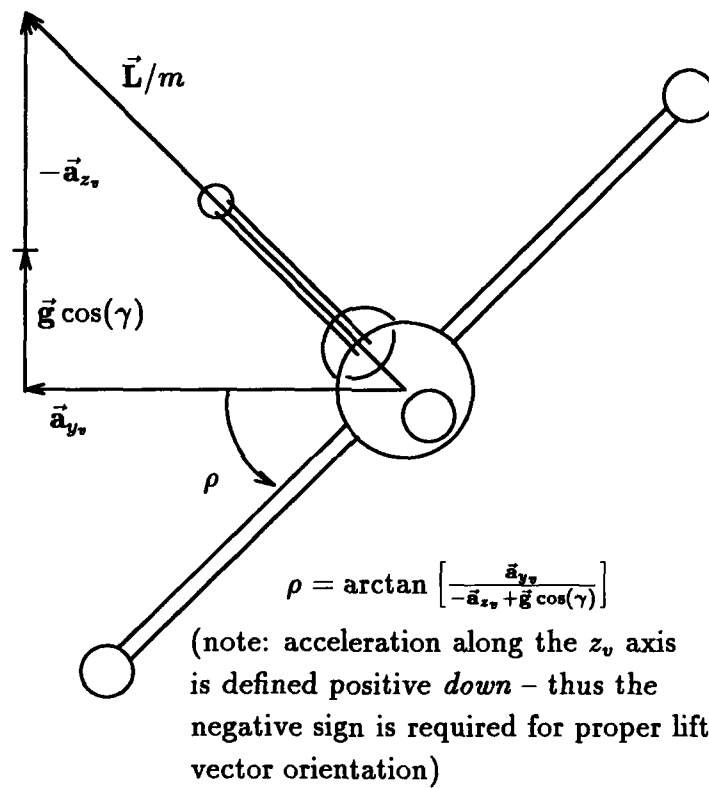


Figure 5.24. Accelerations for Coordinated Turns

will determine roll angle. Sideslip velocity (component of velocity normal to the body frame  $\vec{x}_b - \vec{z}_b$  plane) will be assumed zero. Consistent with the modelling simplifications in the Kendrick and Andrisani efforts [120, 5] and other analyses [104, 143], our atmosphere will be considered at rest with respect to the inertial frame, with on-line corrections to be implemented if target-local wind velocity is available. Note also that, by arbitrary choice of the direction of  $\vec{x}_b$ , we have defined the zero-lift angle of attack ( $\alpha_b$ , as in Chapter IV) to equal zero. Unavoidable real world deviations from these “zero-nominal” assumptions will induce some error into our estimates of aspect angle from kinematics, but we will size our search areas in aspect angle adequately to deal with expected deviations.

For “control configured” classes of aircraft, unconventional assumptions will be made to provide aspect angle as a function of kinematics – for example, an aircraft with “turn-like-a-car” dynamics might be assumed to have only a nominal angle of attack, generally keeping the target body vector aligned with the velocity vector, but rolling in the direction of a turn for pilot comfort and visibility. In our research to date, we have assumed only a conventional aircraft’s coordinated turn motion – note that for any control method in which the plane of the wings is essentially normal to the lift vector, minor deviations from the coordinated turn dynamics result only in an aspect angle position bias error which is ignored by our algorithms. Recall however, as discussed in Sect. 3.6.5, that inherently our recognition algorithms are multiple model estimators – different possibilities or assumptions as to dynamics add a second dimension to the range of hypotheses for which we must compare models to observations, based on discrete choices for the dynamics assumptions. The fundamental dimension in this range of hypotheses is of course described by the different target signatures as functions of aspect angle.

Given some assumption on target body frame orientation relative to the target velocity vector, we need to define the orientation of the body frame relative to the inertial frame. This orientation is provided by the following computation involving direction cosine matrices, which yields the body frame unit vectors in inertial frame coordinates:

$$\mathbf{F}_b = \mathbf{C}_v^b \mathbf{C}_i^v \mathbf{I} = \mathbf{C}_i^b \quad (5.2)$$

where:

$\mathbf{F}_b$  = a three by three matrix, each row of which defines a target body frame unit vector in inertial frame coordinates, and

$$\mathbf{C}_v^b = \begin{bmatrix} \cos(\alpha) & \sin(\rho) \sin(\alpha) & -\cos(\rho) \sin(\alpha) \\ 0 & \cos(\rho) & \sin(\rho) \\ \sin(\alpha) & -\sin(\rho) \cos(\alpha) & \cos(\rho) \cos(\alpha) \end{bmatrix} \quad (5.3)$$

$$\mathbf{C}_i^v = \begin{bmatrix} \cos(\eta) \cos(\gamma) & \sin(\eta) \cos(\gamma) & -\sin(\gamma) \\ -\sin(\eta) & \cos(\eta) & 0 \\ \cos(\eta) \sin(\gamma) & \sin(\eta) \sin(\gamma) & \cos(\gamma) \end{bmatrix} \quad (5.4)$$

and all angles are defined in Figs. 5.23 and 5.24.

Thus, the velocity frame is found relative to the inertial frame by use of the Euler angles identified as heading ( $\eta$ ) and flight path ( $\gamma$ ) angles. The body frame is found relative to the velocity frame by considering (for coordinated turns with zero sideslip angle) the calculated roll angle ( $\rho$ ) and angle of attack ( $\alpha$ ) required as discussed above to generate the required normal acceleration. Other appropriate assumptions are made for unconventionally controlled vehicles. Again, for the truth model or simulated target, these quantities are known – for the motion warping processor, they are estimated.

Given  $\mathbf{C}_i^b$ , the target-to-sensor unit vector is defined in target body frame coordinates as

$$\bar{\mathbf{r}}_{t-s}^b = \mathbf{C}_i^b \frac{\mathbf{r}_{t-s}^i}{|\mathbf{r}_{t-s}^i|} \quad (5.5)$$

where we have simply divided the target-to-sensor vector in inertial coordinates by its magnitude and performed the coordinate transform.

With the target-to-sensor unit vector defined in target body frame coordinates, we now calculate the target aspect angle (azimuth  $\kappa$  and elevation  $\lambda$  of the target-to-sensor vector, see Fig. 5.23) using the following relations (where the superscript  $b_x$ , for example,

means the component of the associated target-sensor unit vector in the target body frame  $\vec{x}_b$  direction):

$$\kappa = \tan^{-1} \left( \frac{\vec{r}_{t-s}^{b_y}}{\vec{r}_{t-s}^{b_x}} \right) \quad (5.6)$$

for positive  $\vec{r}_{t-s}^{b_y}$  and  $\vec{r}_{t-s}^{b_x}$ , or

$$\kappa = \tan^{-1} \left( \frac{\vec{r}_{t-s}^{b_y}}{\vec{r}_{t-s}^{b_x}} \right) + 360^\circ \quad (5.7)$$

for negative  $\vec{r}_{t-s}^{b_y}$  and positive  $\vec{r}_{t-s}^{b_x}$ , or

$$\kappa = \tan^{-1} \left( \frac{\vec{r}_{t-s}^{b_y}}{\vec{r}_{t-s}^{b_x}} \right) + 180^\circ \quad (5.8)$$

for positive or negative  $\vec{r}_{t-s}^{b_y}$  and negative  $\vec{r}_{t-s}^{b_x}$ , and

$$\lambda = \sin^{-1}(-\vec{r}_{t-s}^{b_z}) \quad (5.9)$$

for positive  $\vec{r}_{t-s}^{b_z}$ , or

$$\lambda = \sin^{-1}(\vec{r}_{t-s}^{b_z}) \quad (5.10)$$

for negative  $\vec{r}_{t-s}^{b_z}$  (note that due to SRC model definitions, positive elevation angles are toward the negative  $\vec{z}_b$  direction, as shown in Fig. 5.23).

Using the time derivatives of the matrix  $\mathbf{C}_i^b$  and the target-to-sensor vector in inertial frame coordinates, we can define the time derivative of the target-to-sensor unit vector with respect to the body frame, in target frame angular coordinates. This quantity will define the (nominal) angular rate along the kinematically-computed aspect angle path. The rate of change of the target-to-sensor vector, with respect to the body frame, defined in target body frame coordinates, is given by:

$$\frac{d\mathbf{r}_{t-s}^{b,b}}{dt} = \mathbf{C}_i^b \frac{d\mathbf{r}_{t-s}^{i,i}}{dt} + \dot{\mathbf{C}}_i^b \mathbf{r}_{t-s}^{i,i} \quad (5.11)$$

where:

$\dot{C}_i^b$  = the time derivative of  $C_i^b$ , element by element

and the first superscript on derivatives of  $r$  refers to coordinatization, while the second refers to the frame with respect to which a derivative is taken.

Dividing the above rate by the length of  $r_{t-}^b$ , and taking the component of this normalized rate perpendicular to the unit vector yields the traverse rate, in radians per second, of the aspect angle vector along an instantaneous "great circle" path (of generally constantly changing direction) on the hypothetical aspect angle sphere of unit radius. This traverse rate is now the estimated mean aspect angle rate. If desired, this rate can be isolated into components in azimuth and elevation directions, as defined in Fig. 5.23.

Thus, by the previous sequence of computations, we have estimated the (kinematically *observable*) angular position and rate of  $\vec{r}_{t-}^b$ , (as observed from the target body frame). Recapping, these aspect angle changes are assumed primarily due to (1) pitch control (changing angle of attack) to change the *magnitude* of normal acceleration, or (2) roll control to change the *direction* of normal acceleration, or (3) (less rapid) target motion along curved portions of the trajectory (no change in acceleration relative to the target body frame), or (4) motion of the sensor along the ownship trajectory (the latter assumed known).

Clearly, in the absence of normal acceleration followed in turn by velocity and position changes, target aspect angle changes are *unobservable*, in the estimation theory sense, to our kinematic estimator. The absence of normal acceleration implies that the aircraft is flying an essentially straight (not necessarily level) trajectory, perhaps with a non-zero roll rate. In any case, this class of trajectory is of minor interest to us for two reasons. First, a combatant (as opposed to commercial or military transport) aircraft target seldom flies in a straight line for long periods, unless the pilot is unaware of being tracked (it is generally assumed that an adversary's warning systems will alert him to the presence of active sensors). Second, the fact that the target is flying a straight trajectory implies (unless the target is rolling) that the target-sensor aspect angle is not changing significantly – thus, the high information content *characteristic sequences* that we seek will probably not be available anyway.

However, should we desire to estimate the target's orientation in straight flight, reasonable assumptions can be made. In most cases, the aircraft will be flying with "wings level", i.e., the plane containing the body frame  $\vec{x}_b$  and  $\vec{z}_b$  unit vectors will be perpendicular to the local horizontal plane, and the body frame pitch angle can be estimated from the flight path angle and the angle of attack required to counteract the normal component of gravity. In Sect. 6.2, techniques derived from word spotting in continuous speech with dynamic time warping will be proposed to identify target rolling motion and class in the absence of normal acceleration.

**5.5.3.2 Bounds Around the Kinematic Aspect Path.** The preceding, or first part of this discussion has addressed only the determination of the nominal kinematically-estimated aspect angle path. The ranges or bounds of probable errors in the start and end points of this path, in the angular rates at which the path is traversed, and "off-nominal" errors can be estimated from the statistics of the errors in the kinematic estimates for velocity and acceleration.

The aspect angle position and rate error bounds will in turn define the continuity constraints for dynamic programming sequence comparison methods. With reference to the "gridded" aspect angle region in Fig. 2.9.a, these bounds will define the width of this region, and the length of this region, or "window" to be checked for any *one* target signature.

The problem here is that the relationships between kinematic state estimate errors and aspect angle estimate errors are highly nonlinear in general. For this research, adequate linear approximations were made by defining distributions of starting values about the nominal positions and rates according to the equation:

$$\mathbf{P}_A = \mathbf{EPE}^T \quad (5.12)$$

where:

$\mathbf{P}_A$  = is a first order (linearized) covariance estimate, at any nominal aspect angle along the trajectory, for the error in the angular position and angular rate of the nominal aspect path, in the direction of and normal (cross-track) to the path (a  $4 \times 4$  matrix).

$\mathbf{E}$  = is a 4 (row)  $\times$  6 (column) dimensional Jacobian matrix of partial derivatives, defined by determining the partial derivatives of angular position and rate along and normal (cross-track) to the nominal path (four separate quantities), with respect to the target velocity and acceleration components along each inertial frame axis (a total of six components).

$\mathbf{P}$  = is the (estimated, from an extended Kalman filter and associated smoother) covariance of the target velocity and normal acceleration estimates.

The along-track and cross-track angular error bounds at each measurement observation time allow us to subdivide the total aspect angle region shown in Fig 2.9 into allowable aspect angle regions, or windows, at each time. These windows or subregions will in general overlap, as shown in Fig. 3.2 in Chapter III.

The along-track and cross-track angular *rate* statistics given by  $\mathbf{P}_A$ , then, allow us to *estimate* quantitatively the likelihood of transitions between any two aspect angle states on adjacent windows over a given measurement time interval  $\Delta T = t_{k+1} - t_k$ . Since we have an estimate for the mean aspect angle rate as found in the previous section, we can estimate the natural logarithm  $L_T$  of the Gaussian classical likelihood of the rate implied by a such a transition (less the usual constant term). This in turn can be taken to provide the natural logarithm of the *a priori* aspect angle transition likelihood  $p(\mathbf{x}_{k+1}^a | \mathbf{x}_k^a, \mathbf{Z}_m^d, \omega_i)$ , as discussed in Sect. 3.6 with respect to the Larson and Peschon equations:

$$L_T = [\ln p(\mathbf{x}_{k+1}^a | \mathbf{x}_k^a, \mathbf{Z}_m^d, \omega_i)] - C =$$

$$- 0.5 \left[ \left( \frac{\Delta \mathbf{x}_{AT}^a}{\Delta T} - \hat{\mathbf{r}}_{AT} \right) : \left( \frac{\Delta \mathbf{x}_{CT}^a}{\Delta T} - \hat{\mathbf{r}}_{CT} \right) \right] (\mathbf{P}_{AR})^{-1} \left[ \left( \frac{\Delta \mathbf{x}_{AT}^a}{\Delta T} - \hat{\mathbf{r}}_{AT} \right) : \left( \frac{\Delta \mathbf{x}_{CT}^a}{\Delta T} - \hat{\mathbf{r}}_{CT} \right) \right]^T \quad (5.13)$$

where:

$L_T$  = the log Gaussian classical likelihood (less the usual constant term  $C$ ) of the transition rate implied by movement in aspect angle between two given aspect angle cells

$\mathbf{x}_k^a$  and  $\mathbf{x}_{k+1}^a$  over one measurement interval  $\Delta T$ , for some target class  $\omega_i$ , given  $m$  kinematic measurements  $\mathbf{Z}_m^d$

$\mathbf{P}_{AR}$  = a  $2 \times 2$  matrix, the lower right sub-matrix of  $\mathbf{P}_A$

$\Delta \mathbf{x}_{AT}^a$  = the along-track component of the difference in angle between the two given aspect angle cells  $\mathbf{x}_k^a$  and  $\mathbf{x}_{k+1}^a$  (i.e., the angle component along the direction of the mean expected aspect angle rate)

$\Delta \mathbf{x}_{CT}^a$  = the cross-track component of the difference in angle between the two given aspect angle cells  $\mathbf{x}_k^a$  and  $\mathbf{x}_{k+1}^a$

$\hat{r}_{AT}$  = the mean expected aspect angle rate in the along-track direction, which is found as discussed in the previous section (see Eqn. (5.11) and associated comments)

$\hat{r}_{CT}$  = the mean expected aspect angle rate in a cross-track direction, which is equal to zero by definition

$C$  = the usual factor associated with the natural logarithm of the leading term of a Gaussian probability density, a constant for constant  $\mathbf{P}_{AR}$ .  
and other terms have been defined.

The natural logarithm of the likelihood for the *a priori* starting state  $p(\mathbf{x}_0^a | \mathbf{Z}^d, \omega_i)$  is defined in an analogous fashion, using the upper left  $2 \times 2$  submatrix of  $\mathbf{P}_A$ , corresponding to covariance of angular position about the mean or nominal aspect angle estimate from kinematics at time  $t_0$  (strictly following the Larson and Peschon methodology, this is one time interval prior to the first signature measurement).

It is important to remember that the accuracy of these estimates for angular position and rates and associated transition likelihoods are completely dependent upon the extent to which the kinematic state estimator model matches the actual target behavior. Recall that in the effort discussed in this chapter, we assume that the target acceleration relative to the body frame is *constant* – in other words, a constant turn rate model (not necessarily a *planar* turn). We depend here upon the analysis of kinematic observations (by extended Kalman filter / smoother / curve fit) to ensure that the target is in fact executing a constant acceleration turn as HRR radar signatures are taken. In Chapter VI, we will



combine ideas from Sect. 3.4 and Chapter IV to free us from dependence on the smoother and constant acceleration turn model, as previewed in Sect. 3.9.

Straightforward extensions of Eqn. (5.12) allow for calculation of angular position and rate error "covariances" due to filter state errors *and* errors in assumption of other variables such as coefficient of lift  $C_{L\alpha}$  (normally assumed at some nominal), sideslip angle (normally assumed zero), as well as off-nominal path errors and error rates. Note that this technique can also be used to estimate a (pseudo-) measurement error "covariance" for the aspect angle pseudo-measurement provided to the aspect angle Kalman filter in the Kendrick estimator (this value was a fixed input in the Kendrick research).

These careful computations of aspect angle position and rate error may not seem worth the effort, but they yield a great deal of interesting information. They show, for example, that the usual assumption of "square" [164] or "circular" [20] aspect angle windows for searches on aircraft target models is probably suboptimal when target kinematics are reasonably well known – generally, the window extent should be greater in the direction of possible angle errors due to target *roll* than it is in the direction of angle errors due to changes in *pitch* or angle of attack. Since computational loading is driven by window size (and the discretization fineness or granularity of aspect angle cells in the window) optimal sizing of aspect angle windows could be a significant issue.

For most of the trajectories in this research, the "along-track" direction corresponds to changes in pitch, and the "cross-track" direction corresponds to changes in roll. Accordingly, a typical aspect angle window size in this research was 10 degrees in the along-track direction by 20 degrees in the cross-track direction. For a two-g turn tracking scenario like that for Fig. 5.27 and others to be discussed in Sect. 5.6, this corresponds to approximately  $\pm$  two standard deviations of angular position error for both along-track and cross-track aspect angle position, as estimated from kinematics, or correspondingly more standard deviations of error for higher acceleration turns (these angular standard deviations are the square roots of the first two diagonal elements of the matrix  $\mathbf{P}_A$  above). Simply, the harder the aircraft turns, the more one can be sure of its aspect angle.

*5.5.3.3 Ground Targets.* Derivation of kinematically-estimated aspect angle and aspect angle rate for ground vehicles is quite analogous to the above, except that the coordinated turn assumption (or other assumption as appropriate) is replaced by assumptions that the vehicle is generally pointed in the direction of its velocity vector, and that turns occur on a flat surface, or that the surface at the vehicle location is reasonably well known (as from stored map data).

*5.5.4 Extracting the Kinematically-Estimated Feature Observable Sequence.* As shown in Fig. 3.2, we have used kinematic measurements to define, for each potential target class, a set of aspect angles or "map" over which we will make comparisons with observed signature measurements. The next step is to load or associate the map "cells" with library signatures from the database corresponding to the desired target model.

There are any number of more or less efficient ways to approach this software engineering issue. In this research, the map is simply a Fortran array which can be loaded with aspect angle values, and the library signature value closest to each map aspect angle is found by testing each library signature's aspect angle against each map value as the program reads the entire library signature file. If a library signature within one degree in aspect and ten degrees of polarization is not found for each map cell during the library read operation, an error advisory prints to output.

For actual on-line operation, a much more efficient method would be to "point" or reference the map location to the desired library element. That was not done for this research to avoid keeping all of the libraries loaded in storage at all times. This and subsequent processes can be performed identically for ground and air targets.

*5.5.5 Comparing the Measured Sequence to the Candidate Target Signatures.* At this point we have a sequence of actual signature observations from some "true" target, and an aspect angle "map" loaded with appropriate signatures for some candidate target class. The map is subdivided into aspect angle windows, or search bounds corresponding to each signature measurement time. We wish to compare the observed signatures to the map signatures and quantify the maximum joint likelihood (or a comparable measure) that the

observed signatures came from the candidate target class. A number of algorithms have been investigated for this task.

Those algorithms to follow which employ dynamic programming are implemented in simple code structures of the following form (see [176:App. B]). Note that computations for the many aspect angle cell / target model combinations can and should be done in parallel for fast on-line execution, but that was not implemented in our simulations.

- (1) (Loop 1) For each window (equivalently, the current signature):
- (2) (Loop 2) For each aspect angle cell in the current window (the current cell):
- (3) Compute likelihood or cost of generating the current signature from this cell.
- (4) Compute allowable predecessor cells (none for first window).
- (5) (Loop 3) For each allowable predecessor cell:
- (6) Compute joint sequence likelihood to current cell, or other measure of path cost.
- (7) End Loop 3.
- (8) Select allowable predecessor cell giving maximum joint likelihood or minimum total cost to current cell, and store this value, predecessor cell location (pointer) and other data as required, referenced to the current cell location.
- (9) End Loops 1 and 2.
- (10) Select maximum joint likelihood / minimum total cost cell in final or latest window and retrace pointers to find best path.

We now discuss the individual algorithms compared in this research. The discussion to follow assumes the use of classical likelihood functions or related quantities (e.g., Mahalanobis metrics) for observed signature-to-map signature comparisons.

*5.5.5.1 The Independent Look (IL) Algorithm.* The IL Algorithm is a conventional "independent look" decision theoretic target recognizer as discussed in Sect. 2.2. No restriction is placed on the resulting "pose estimate" sequence – we find  $\text{MAX} \ln[p(\mathbf{z}_k' | \mathbf{x}_{k,n}^a, \omega_i)]$  within the specified aspect angle window at each measurement time  $t_k$ . These values are summed incrementally at each event to find the natural logarithm

of the maximum joint likelihood. With reference to Fig. 3.2, this means that the recognizer is free to choose any sequence of aspect angles on the map, so long as each aspect angle choice falls within the proper window.

**5.5.5.2 The Perfect Knowledge of Aspect (PKA) Algorithm.** The PKA Algorithm provides an upper bound on recognition performance in that it assumes that the recognizer *knows perfectly* the true aspect angle over time for each target class executing the observed maneuver. For the Mahalanobis metric, this figure is simply the natural logarithm of the joint maximum (classical) likelihood for *known* or true aspect angle  $\mathbf{x}_{t,n}^a$  over time,  $\ln \{ \prod_{n=1}^k [p(\mathbf{z}_n^f | \mathbf{x}_{t,n}^a, \omega_i)] \}$ . These values are summed incrementally at each event to find the natural logarithm of the joint likelihood. Note that if the kinematic aspect angle estimate is poor, or aspect angle error bounds are too low, the sequence of true aspect angles may *not* even fall on the map in Fig. 3.2, even for the true target.

**5.5.5.3 The Fixed Bound (FB) Algorithm.** The FB Algorithm is an implementation of the Le Chevalier algorithm, as that approach is believed to work: the algorithm has no information from kinematics on the expected direction of aspect angle change, but knows that the change is bounded. No subsequent processing is applied. This algorithm was illustrated in Fig. 3.11. The algorithm finds the natural log of the term in Eqn. (5.14) (a modification of Eqn. (3.11)) as discussed in Sect. 3.6. Note that the term  $p(\mathbf{Z}_k^f | \hat{\mathbf{X}}_{fb,k}^a, \omega_i)$  means specifically that this term is the joint classical likelihood of the observed  $\mathbf{Z}_k^f$ , conditioned on their having been associated with the aspect angle sequence  $\hat{\mathbf{X}}_{fb,k}^a$  found by the fixed bound process on model  $\omega_i$ .

$$p(\mathbf{Z}_k^f | \hat{\mathbf{X}}_{fb,k}^a, \omega_i) =$$

$$\left\{ \max_{\mathbf{x}_{k,n}^a} I^*(\mathbf{x}_{k,n}^a, k | \omega_i) \right\} = \left\{ \max_{\mathbf{x}_{k,n}^a} \left[ p(\mathbf{z}_k^f | \mathbf{x}_k^a, \omega_i) I^*(\mathbf{x}_{k-1,n}^a, k-1 | \omega_i) \right] \right\} \quad (5.14)$$

**5.5.5.4 The Larson and Peschon (L&P) Algorithm.** The L&P Algorithm finds the natural log of the term in Eqn. (3.11). Note that the numerical values of this algorithm include contributions due to the *a priori* aspect angle transition likelihood

$p(\mathbf{x}_{k+1,n}^a \mid \mathbf{x}_{k,n}^a, \mathbf{Z}_m^d, \omega_i)$ . Contributions associated with the L&P “*a priori*” state  $\mathbf{x}_{0,n}^a$  were not included, to provide for unbiased comparison with the results of other algorithms (none of which include such a term, although they could in theory).

The process involved in calculating the likelihood of a given path was illustrated in Figs. 3.3 and 3.4 of the last chapter. Computations for this algorithm proceed exactly as for the FB algorithm, except that log transition likelihoods are summed along with log likelihoods of observed signature generation.

#### 5.5.5.5 Classical Sequence Comparison – One Dimensional (1-D Warp).

The 1-D Warp Algorithm defines contiguous one-dimensional paths in aspect angle, parallel to and including the “nominal” path given by the extended Kalman filter/smoothing kinematic estimate. An algorithm functionally identical to one-dimensional, unrestricted endpoint [176, 182] dynamic time warping is performed along each trajectory, as discussed in Sects. 2.4.2 and 3.7.2. The basic form requires contiguous matching – every aspect angle cell along a given path must be matched to a measurement. In this research, each local path cost is normalized by the total number of associations along that path, and the path with minimum normalized cost at the final (or latest) aspect angle window is selected as the best path. The output likelihood value can be taken from this normalized cost, or it can be taken from the “best” individual match for each signature along the selected contiguous path (the latter choice is used in the results shown below, to achieve the minimum cost or maximum likelihood on the true model, for better comparison with the other algorithms).

Best path selection based on normalized cost is a departure from usual DTW practice, and can lead to violations of the “Principle of Optimality” [23], but worked well in our tests, since for the proper match of measurements to target class, and constant measurement noise statistics across the target length, the local average matching cost is expected to be near the global average. Without such normalization, however, *in the scenarios used here*, conventional (“contiguous”) sequence warping often fails to follow an optimal path on the true (unknown *a priori*) target model (i.e., a path close to the origin locations of the true signatures). This failure in turn causes a poor reference point for comparison to matches on the incorrect models.

The desire for local path normalization is driven by a particular combination of factors in the scenarios investigated here. In a typical case for *this* research, we may have six observations taken at 0.4 second intervals, giving an elapsed time of 2.4 seconds. Now over this same period, an aircraft target executing a constant 4-g turn in some horizontal plane at a speed of 800 meters per second will exhibit a 22.0 degree aspect angle change to a stationary observer in the same plane. Assuming a 1.0 degree granularity signature map, and some overlap (say 5 degrees) at each end of the aspect angle "map" to allow for bias uncertainties in the target aspect angle, we may have  $22 + 2(5) = 32$  horizontal cells in our aspect angle map. Thus, the ratio of elements in the two sequences to be compared is 6/32.

If the correct association proceeds in an essentially linear fashion (6 observations matching to 22 map cells), we expect that each observation will associate with approximately 4 map elements. This 6/22 ratio is much less than the 1/3 ratio that previous researchers in speech processing found to be the minimum acceptable [168, 181]. The practical effect of this small ratio is that, without some form of compensation, the six observations will match to considerably fewer of the map elements than 22. For the *true* target, this will result in a match that is non-optimal in a practical sense – providing signature likelihoods that are a poor basis for comparison with results from incorrect target model associations (of course, the true target is unknown *a priori*). The probability of correct recognition will be very low.

The problem is that the classical dynamic programming equations given in Sect. 2.4.2 lead toward a bias for shorter paths in the lattice of Fig. 2.8. Previous researchers have addressed this issue by a wide variety of continuity constraints that reduce the penalty for longer or "non-diagonal" paths, such as the method discussed in Sect. 5.4.1. Several researchers have pointed out that the objective is a path-length normalized warping cost, but have maintained that this normalization cannot be made locally at each step in the dynamic programming process, since the minimization process is made locally based on cost for each path only [168].

In this research, however, the author has done just exactly that – normalized locally, based on the total cost *and* number of associations in each candidate predecessor path.

The point here is that dynamic programming does *not* restrict the user to make local decisions based only on cost to a given point [71:12]. Information other than path cost can be *retained* and used for later decisions. That being said, the concept of performing an addition and a division for each cost calculation is a significant deviation from usual dynamic programming practice, and not one to be used lightly. However, represented in usual dynamic programming form as a sum of costs at each stage, path normalization can be considered simply as a case in which incremental costs at each stage are a (complicated) function of the path to that point, and philosophically, dynamic programming practice provides for this case.

Referring back to Eqn. (2.29), the process of path cost computation with local length normalization can be written (and executed in the otherwise usual fashion) as shown below in Eqn. (5.15): this is essentially Eqn. (15) of [168].

$$D_n(C_k) = \frac{d(c_k) + (k-1)\text{MIN}[D_n(C_{k-1})]}{k} \quad (5.15)$$

where:

$c_k = [a_j, b_l]$  is the  $k$ -th element in a sequence of allowable associations of elements from sequence  $A$  with elements of sequence  $B$ , this particular association being between element  $a_j$  and element  $b_l$

$C_k = \{c_1, c_2, c_3, \dots, c_k\}$ , the minimum normalized cost sequence of associations leading to and including association  $c_k$

$d(c_k)$  = the cost or distance of association  $c_k$ , i.e., the distance in some metric between element  $a_j$  and element  $b_l$

$D_n(C_k)$  = the normalized cost of reaching and accomplishing association  $c_k$  by the minimum normalized cost sequence of allowable associations

$D(C_k) = kD_n(C_k)$  = the total cost of reaching and accomplishing association  $c_k$  by the minimum normalized cost sequence of allowable associations

The key concept to keep in mind is the "Principle of Optimality", which, reworded somewhat from the discussion in Sect. 2.4, says that dynamic programming will yield

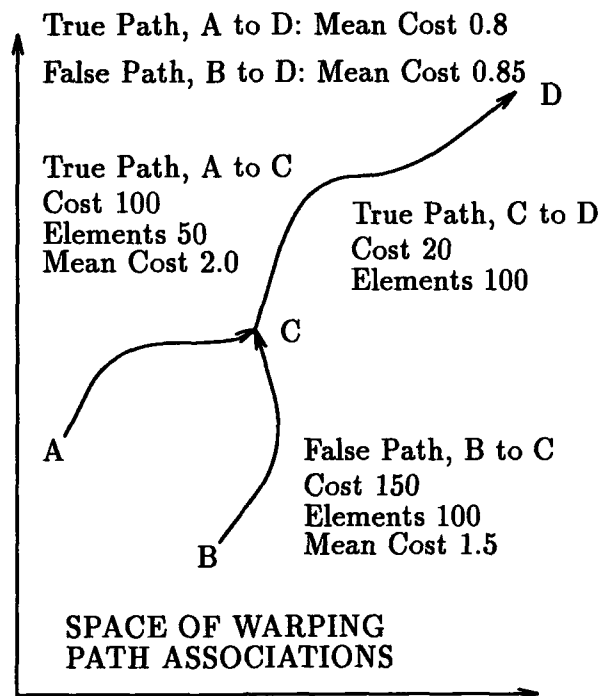


Figure 5.25. Violation of the Principle of Optimality From Local Path Normalization

the optimal path, when the locally optimal path (during the matching process) lies on the globally optimal path. Now, for a local normalization process, it is very simple to construct a set of conditions in which the locally optimal (minimum average) path does *not* lie on the globally optimal path. Fig. 5.25 shows such a case – note in the figure that at point C, a dynamic programming algorithm using normalized path cost would select the “false” path from B to C as the predecessor route, while in fact the “true” path in a globally optimal sense is from A to C, and on to point D. Thus, local normalization can lead to violations of the Principle of Optimality.

However, in the particular application discussed here, we expect that in a particularly important case, the local average path cost will equal the global average path cost. This occurs when three conditions are satisfied: (1) the observations and library map correspond to the same target model, (2) the aspect angle transition rate (from kinematics, generally) used to define the map is close to the true one which generated the observations, and (3) the signature noise process statistics are constant over the aspect angles and times of interest.



Now condition (1) will apply when we are making a correct target-model association, for a well-modelled target. Condition (2) applies when we have a good kinematic estimate of aspect angle rate. Finally, as noted in Sect. 5.3.4, Condition (3) appears to apply to HRR radar signatures in particular and perhaps others in general.

In other words, when making a correct target-to-model association with good tracking information, local path normalization may not change our answers significantly. The benefit of normalization is that it allows us to demonstrate a "contiguous" path matching concept in this scenario – the disadvantage of normalization is the computational cost of the division operations. As noted above, normalization is essential to allow the algorithm to make a correct match on the true (unknown *a priori*) target model. What happens for incorrect target associations is irrelevant, as long as the matching path cost is higher than for the true target model.

In any case, we must observe that in speech processing, obedience to classical rules for dynamic programming sequence comparison frequently leads to matches that are not "optimal" in a practical sense [168]. This is where the "art" in the "art and theory of dynamic programming" [71] has to be applied [181]. The author's use of local path normalization is such an example, and it worked as intended, as we shall see in Sect. 5.6.

Closing this discussion on path normalization, it is important to emphasize that the desire for normalization was only driven in this case by the high ratio of library aspect angle cells to measurements. This ratio is a function of the desired discretization of aspect angle, the signature sampling interval, and the target turn rate. Under conditions where the ratio of library aspect cells to measurements is lower, normalization will not be required, as it is not for conventional applications.

Finally, recall that in Sect. 3.8.2, it was noted that conventional sequence comparison may have a potential for reduced computational requirements in comparison with an algorithm with Larson and Peschon-type continuity constraints, since conventional sequence comparison algorithms tend to limit the number of predecessor points severely. In the implementation discussed here, this potential is reduced by the need to normalize, or divide by the total number of associations along each path.

#### 5.5.5.6 Classical Sequence Comparison – Two Dimensional (2-D Warp).

The 2-D Warp Algorithm uses the same set of trajectories defined for the one-dimensional case, but local continuity constraints allow the optimum path to move from one trajectory to its right or left neighbor (see Fig. 3.8). Other factors are as for the one-dimensional case.

Note that for any given aspect angle cell in any window, we must now consider up to six possible predecessor paths – three from adjacent cells in the same window and three from cells in the previous window. The dotted line labeled “2-D” in Fig. 3.8 showed how a particular association trajectory might proceed. The path length normalization process is just as in the previous section. Thus, we have increased the dimensionality of the problem, but allowed the algorithm to respond to “off-nominal” path conditions where the true path is not parallel to the nominal or kinematic path.

Like other “tuning” issues associated with all of these dynamic programming sequence comparison algorithms, this increased “flexibility” has pros and cons, as implied in Sect. 3.7.2. From the *pro* perspective, loosening kinematic restrictions on signature matching will help an algorithm to find the best match on the *true* (unknown *a priori*) target model, if the kinematic state estimate, or basis for restrictions, was significantly incorrect. From the *con* perspective, loosening kinematic restrictions will tend to allow an algorithm to find an improperly close match on an *incorrect* target model. Limiting the latter effect, of course, was precisely our original objective.

In any case, where kinematic information provides an accurate estimate of the direction of aspect angle change, we expect that the one dimensional algorithm will outperform the two dimensional one – the two algorithms will provide essentially the same answer for the true (unknown *a priori*) target model, but the flexibility of the two dimensional algorithm will allow it to find an undesirably close match on some *incorrect* target model. Conversely, if the true and kinematically-estimated aspect angle paths cross (are not parallel), the two dimensional algorithm will have a better chance of “finding home” on the true target model – the lower cost it achieves here may offset any penalties incurred by finding an improperly close match on an incorrect target model.

**5.5.6 Conclusion.** The material in this section has covered the operation of the dynamic programming-based sequence comparison techniques and conventional or ideal algorithms against which they will be evaluated. It should be clear that the various algorithms have competing advantages and disadvantages with respect to optimality, computational burden, etc. In the next section, we will evaluate the relative performance of these algorithms, using both conventional performance measures and the generalized ambiguity function, as discussed in Sect. 3.11.

## **5.6 Research Results and Discussion**

**5.6.1 Overview.** The following subsections discuss key observations regarding the results of simulations constructed as outlined in the previous part of this chapter. Additional equations, results, and observations regarding the operation of the kinematic smoother *per se* are given in App. C.

**5.6.2 Kinematic State Estimation.** For this research, the nine state extended Kalman filter discussed in Sect. 5.5.2 was implemented using the "Multimode Simulation for Optimal Filter Evaluation" (MSOFE) software package [46]. The smoother and target recognition algorithms were implemented separately using a file structure developed from that which links MSOFE with the associated plot postprocessing routine, "MPLOT".

Fixed lag smoother "lags" of two to three seconds were found to be entirely adequate – estimates of aspect angle and rate were established with adequate quality to permit the sequence comparison algorithms to work as expected. Fig. 5.26 shows extended Kalman filter and smoother performance in estimating one inertial component of target acceleration over 20 runs, where the true target acceleration is two g's ( $64 \text{ ft/sec}^2$ ) during the period from three to eleven seconds and zero elsewhere. The upper solid curve is mean extended Kalman filter error (similarly bounded on either side by curves for the mean extended Kalman filter error  $\pm$  one standard deviation), while the lower solid curve is mean smoother error (bounded on either side by curves for the mean smoother error  $\pm$  one standard deviation). The smoother delay here is 2.0 seconds – the smoothed estimate (curve number two) is available from 2.0 to 11.9 seconds, using filter information out to 13.9 seconds (the filter

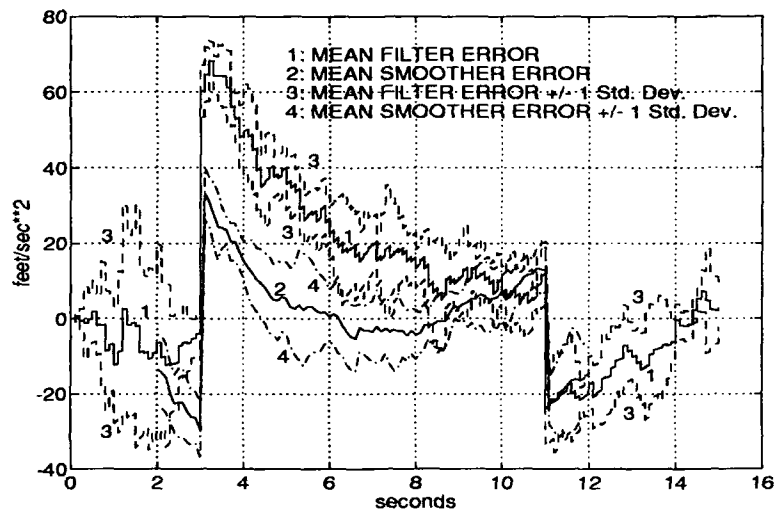


Figure 5.26. Acceleration Estimation Error: Mean +/- One Standard Deviation

runs on until 15.0 seconds). Note that the filter starts with true *and* estimated acceleration equal to zero – hence the zero error at the start of the run.

As noted earlier, although the optimal smoother did an excellent job of correcting extended Kalman filter state estimates as shown here, acceleration estimates were still too noisy from measurement event to measurement event to provide smooth aspect angle estimates, particularly in state directions where insufficient true acceleration made use of the optimal smoother pointless [154:11]. Also, the smoother-derived acceleration estimates were rather more noisy than desired for identifying steady-state acceleration conditions, which indicate target turn events for recognition. Nominally, a steady-state turn event was identified when estimated target acceleration in the target body frame remained within  $\pm 5 \text{ ft/sec}^2$  of a running average over a one-second period. To meet these stringent smoothness conditions, second degree polynomials were fitted to the filter/smooter position estimates, and differentiated twice to obtain an acceleration estimate with error magnitudes that closely follow the mean smoother value in Fig. 5.26. Additional plots showing filter, smoother, and curve fit results are provided in App. C.

The disadvantages of smoothing are added processing and the fact that our target information is no longer real time. In general, we found that a high quality ( $\pm 20\%$ ) estimate of the target acceleration was obtained with a four-second delay – two seconds for the fixed-lag smoother and two seconds for polynomial curve fitting. Following onset of a major maneuver, 2-3 more seconds of delay are desirable to identify steady state conditions (note how the fixed lag smoother mean error curve in Fig. 5.26 begins to level out near the five second point). In any case, as shown in Fig. 5.26, for a 2-g turn lasting as little as eight seconds, the target acceleration can be estimated with high confidence for approximately five seconds (i.e., from approximately the 5 second point to the 10 second point). As we will show below, the advantage accrued in better state estimates can be well worth the wait and processing, particularly for turning accelerations in excess of 1 g.

Once the target velocity and acceleration states are estimated and assumed to be in steady state relative to the target body frame, calculation of target-sensor aspect angle and aspect angle rate are straightforward for any set of assumptions on target control parameters, as shown in Sect. 5.5.3. Even for the relatively large kinematic measurement errors modelled in this research, after smoothing and curve fitting, kinematically-based aspect angle estimates were never observed to lie more than ten degrees from the true (coordinated turn) figure. Aspect angle rates were never more than 25% in error. Even with high quality kinematic measurements, aspect angles defined from filtering *without smoothing* were often subject to much larger errors, particularly at the onset of turn events.

For example, note in Fig. 5.26 that, at the 4-second point, the mean error in the filter-estimated acceleration is approximately  $50 \text{ ft/sec}^2$ , while the mean error in the smoother-estimated acceleration is only about  $15 \text{ ft/sec}^2$ . Therefore, depending on the sensor position, an aspect angle estimate based on the corresponding mean filter acceleration estimate could be in error by as much as 40 degrees – the difference between roll angles required to achieve (1)  $64.4(\text{true}) - 50(\text{error}) \approx 15 \text{ ft/sec}^2$  and (2)  $64.4 \text{ ft/sec}^2$  (two g's, true) horizontal acceleration planar turns. In comparison, a smoother-derived aspect angle estimate based on the corresponding mean smoother acceleration estimate of  $64.4(\text{true}) - 15(\text{error}) \approx 50 \text{ ft/sec}^2$  would be in error by only about 6 degrees. (From the equation in Fig. 5.24, for a lateral acceleration estimate of  $50 \text{ ft/sec}^2$  in a level, planar turn, we obtain a

roll angle of  $\arctan(50/32.2) = 57.38$  deg, versus the figure of  $\arctan(64.4/32.2) = 63.43$  deg for the actual two-g lateral acceleration.)

**5.6.3 Recognition Algorithm Performance.** Typical generalized ambiguity function (GAF) outputs obtained in this research are shown in Figs. 5.27 and 5.28. The dotted vertical lines on each figure indicate parameter (target) interpolation values for which likelihood functions were defined, and spline curve fits connect the mean function values to provide the curves shown. Relevant target trajectory parameters are shown in each figure. As in Chapter IV, the nominal sensor-to-target range is 100,000 feet.

Note in each set of generalized ambiguity functions that the Independent Look (IL) algorithm defines the upper bound on performance (worst), and the Perfect Knowledge of Aspect (PKA) algorithm defines the lower bound (best, but unattainable in practice). The Fixed Bound (FB) algorithms provide significantly improved separation from the IL result, but the algorithms which fuse filter/smoothing-provided *observed* kinematic information provide separation equal to or better than that of the FB algorithm in each of these cases. Note that the fixed bound extent is increased in Fig. 5.28 to allow the algorithm to follow the higher turn rate of this scenario. In general, with good kinematic estimates of aspect angle state, the matching algorithm performance improves as the level of kinematic restriction increases. Thus, the order of improving performance is expected to be IL, FB, 2-D Warp, 1-D Warp, L&P, and PKA.

Observe the highly nonlinear shape of the GAFs in Fig. 5.28 (and of the PKA GAF in Fig. 5.27). Our results have shown that interactions between the HRR radar sweep-to-sweep comparison metric and the changes in the parameter space due to the morphing process can create apparently anomalous results – e.g., cases where the measurements from an F-4 were closer in Mahalanobis metric sense to sweeps from the MIG than they were to sweeps yielded by an interpolated target only 25% removed from the F-4. These cases resulted from the relative motion of scatterers during the morphing process, and were found to be physically reasonable after investigation. Modified morphing rules can resolve these anomalies if desired – transitions in parameter space “between” two target parameter

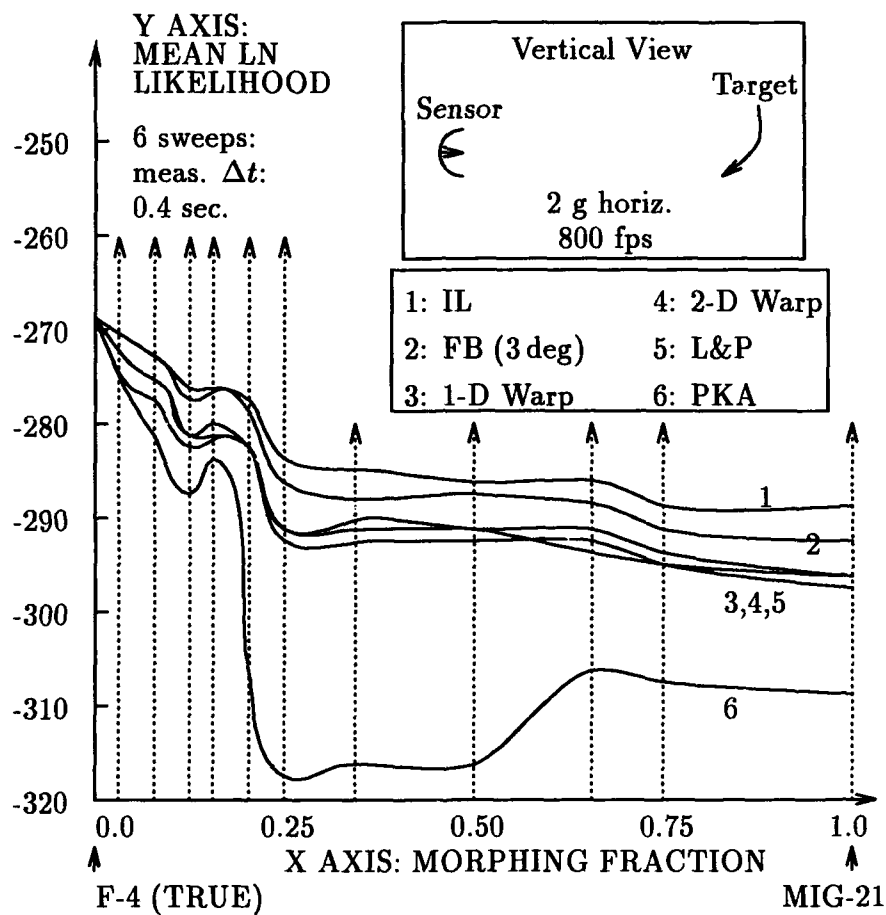


Figure 5.27. Generalized Ambiguity Function for Case 1. Mahalanobis metric, 6 dBsm std. dev. noise.

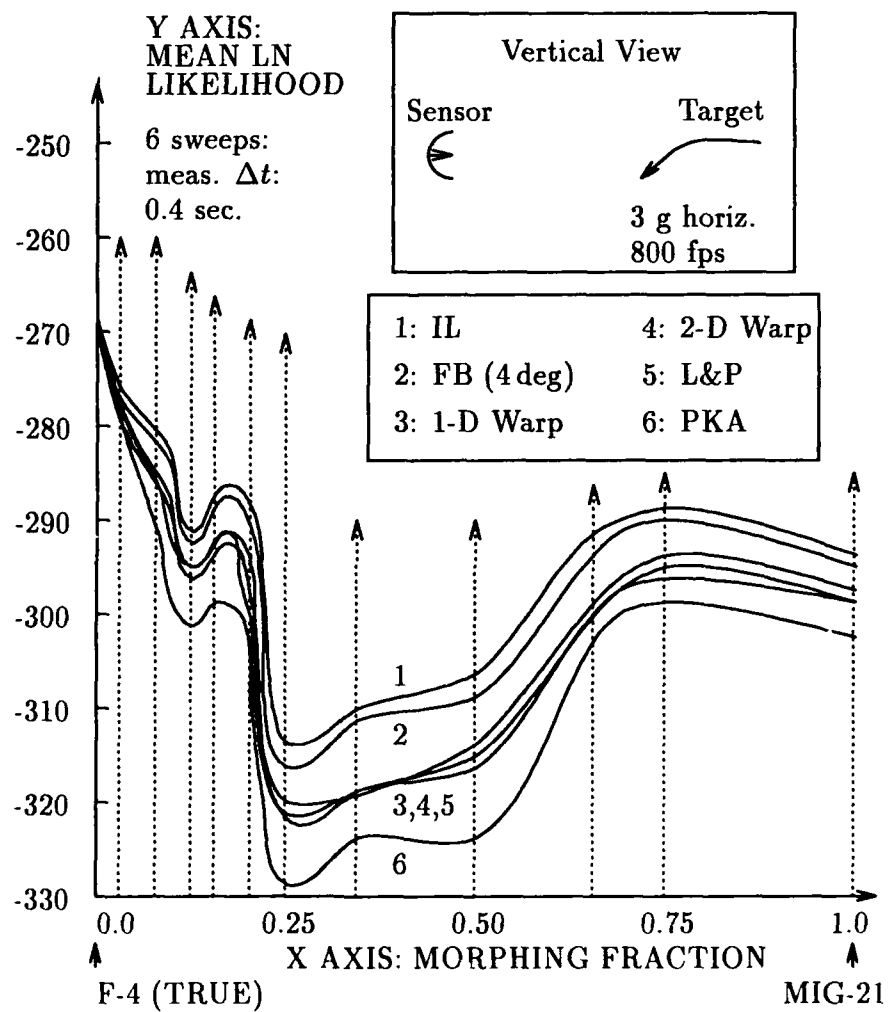


Figure 5.28. Generalized Ambiguity Function for Case 2. Mahalanobis metric, 6 dBsm std. dev. noise.



sets or locations can be made along *any* number of paths to yield *any* number of different generalized ambiguity function shapes.

Although not shown here, the moments of the distributions about the mean values giving the GAF are small (e.g., standard deviations of 3-4 units). Thus, the parent target classes are readily separable *in the feature space (simulated HRR radar) and metric (Mahalanobis) used here* with any of the algorithms shown – the intent here is to show *better* separation with fusion of motion information. This effect is particularly noticeable for pseudo-targets close to the parent (F-4) which provides the measurements. In this region, the FB algorithm provides little improvement over an IL algorithm.

In fact, as discussed in Sect. 5.3.5, considerable effort was required in this research to define target models, trajectories, and noise levels such that *any* algorithm would fail to indicate the correct target. For any pair of targets, trajectory, and kinematic/signature noise realization, failure of an algorithm is defined as the algorithm indicating a higher likelihood (of generating the observed signatures) for an *incorrect* target model than for the *correct* (unknown *a priori*) target model. The scenario that ultimately provided the desired stressing performance was a combination of a trajectory like that in Figs. 4.1 and 5.27 with scatterer-augmented targets and high noise (9 dBsm std. dev.), producing signatures like those in Figs. 5.17 through 5.19. Fig. 5.29 expresses the results of this scenario in classical terms of percent correct recognition, in a format similar to that used for the generalized ambiguity function curves. Note that the PKA algorithm provided 100% correct recognition in every case, and so is not shown explicitly.

Fig. 5.29 shows the percentage of correct recognition for the six algorithms of interest, defined for target parameter (morph) values of 0.04 to 0.25, where morph value 0.0 (the true target) is the scatterer-augmented MIG-21, and morph value 1.0 is the scatterer-augmented and down-scaled SU-22. Clearly, for pseudo-targets at small morph fractions, with one exception, the sequence comparison techniques out-perform the Independent Look (IL) algorithm. The "optimum" Larson and Peschon technique does particularly well, because its explicit aspect angle transition likelihoods do the best job, relative to other algorithms, of restricting it from finding an improperly high likelihood match on an incorrect model.

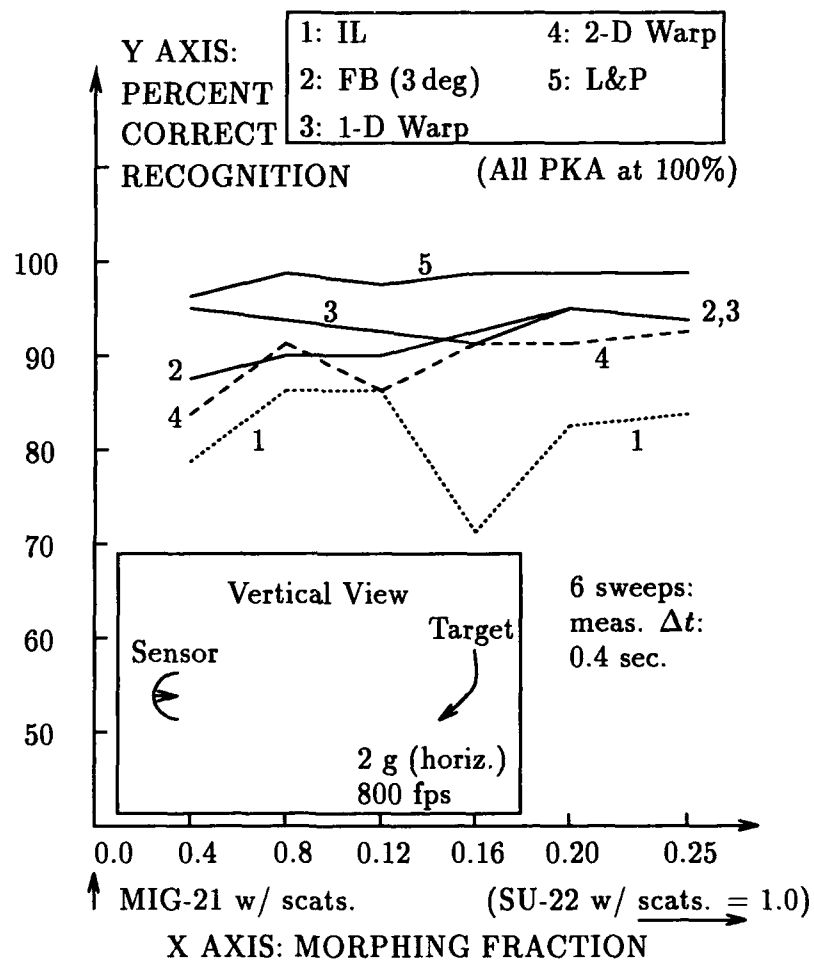


Figure 5.29. Percent Correct Recognition for Case 3. Mahalanobis metric, 9 dBsm std. dev. noise.

Fig. 5.30 shows the generalized ambiguity functions corresponding to Fig. 5.29. The increased ambiguity is evident in the much more closely grouped generalized ambiguity function values for this case.

Since the generalized ambiguity function plots shown thus far have placed the *true* target or parameter set on the left boundary, they are ill-defined to demonstrate the curvature around that parameter point which we associated with the Cramér-Rao lower bound (CRLB) in Sect.2.7. Accordingly, the same set of targets and algorithms was used in a Monte Carlo set for which the "12% Morph" is the true target, or origin of measurements. The generalized ambiguity function plot resulting from this case is shown in Fig. 5.31. Note that the increase in curvature around the true target location is clearly evident as we proceed from the Independent Look algorithm to the Perfect Knowledge of Aspect algorithm. This increased curvature implies progressively lower Cramér-Rao bounds on the parameter estimates for the respective recognition algorithms, bounded by that for the PKA algorithm, which is unattainable in practice.

Figure 5.32 demonstrates that the high ambiguity evidenced in the previous case was in large part a function of trajectory or aspect angle, since that scenario caused the target scatterers to be highly "concentrated" in range. In Fig. 5.32, we have used the same, scatterer-augmented target set with a trajectory in which the sensor-to-target vector is essentially along the target longitudinal axis, and the scatterers are separated in range. The target signatures are now sufficiently distinct from one another that the percentage of correct recognition is again 100% for all algorithms (therefore, no "percent correct" plot is given), and the generalized ambiguity functions show much higher variation from left to right.

In Figure 5.33, we return to a stressing trajectory similar to that of Fig. 5.30, but with a non-planar turn and models that are fundamentally more distinct in structure – a YAK-28 and a B-737 scaled to the approximate size of the YAK (this B-737 model is comparable in many respects to a small business class jet). For the morph fractions investigated, we again see the usual trends for reduction in ambiguity. The targets are again sufficiently distinct that the percentage of correct recognition is 100% in each case, and no plot of those results is required.

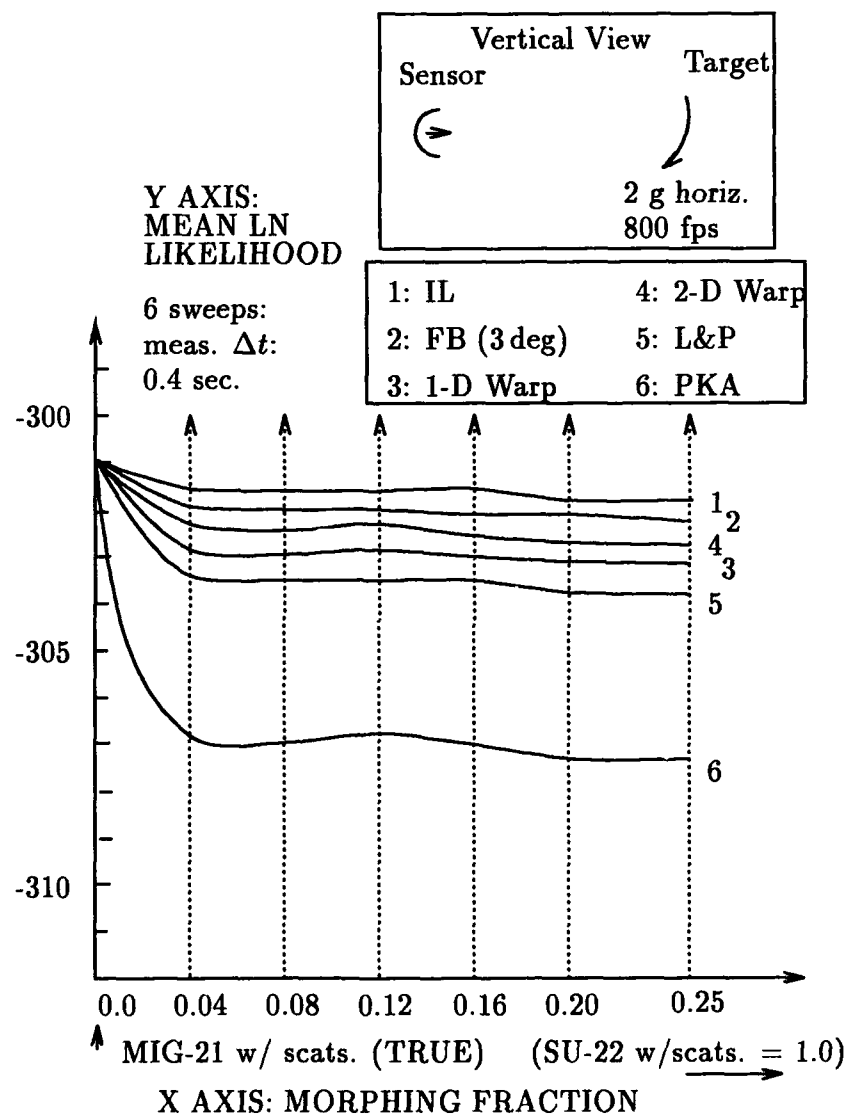


Figure 5.30. Generalized Ambiguity Function for Case 3. Mahalanobis metric, 9 dBsm std. dev. noise.

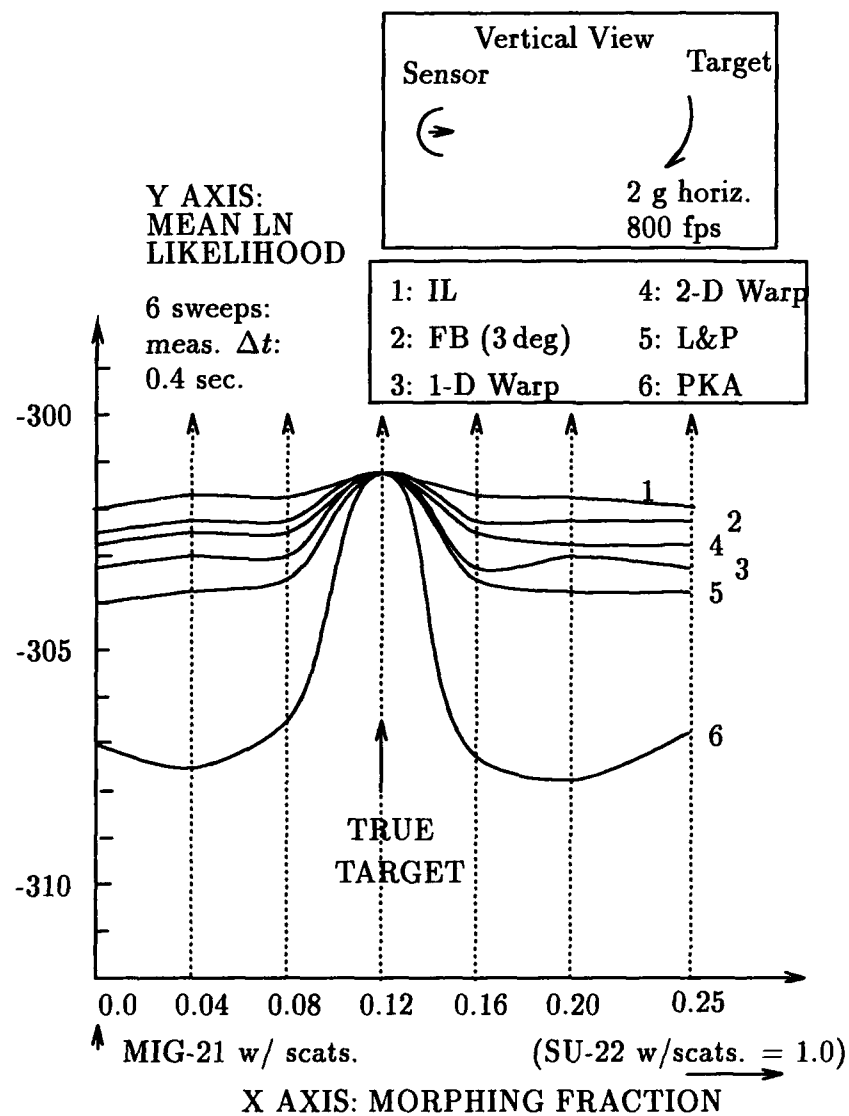


Figure 5.31. Generalized Ambiguity Function for Case 3 (Modified). Mahalanobis metric, 9 dBsm std. dev. noise.

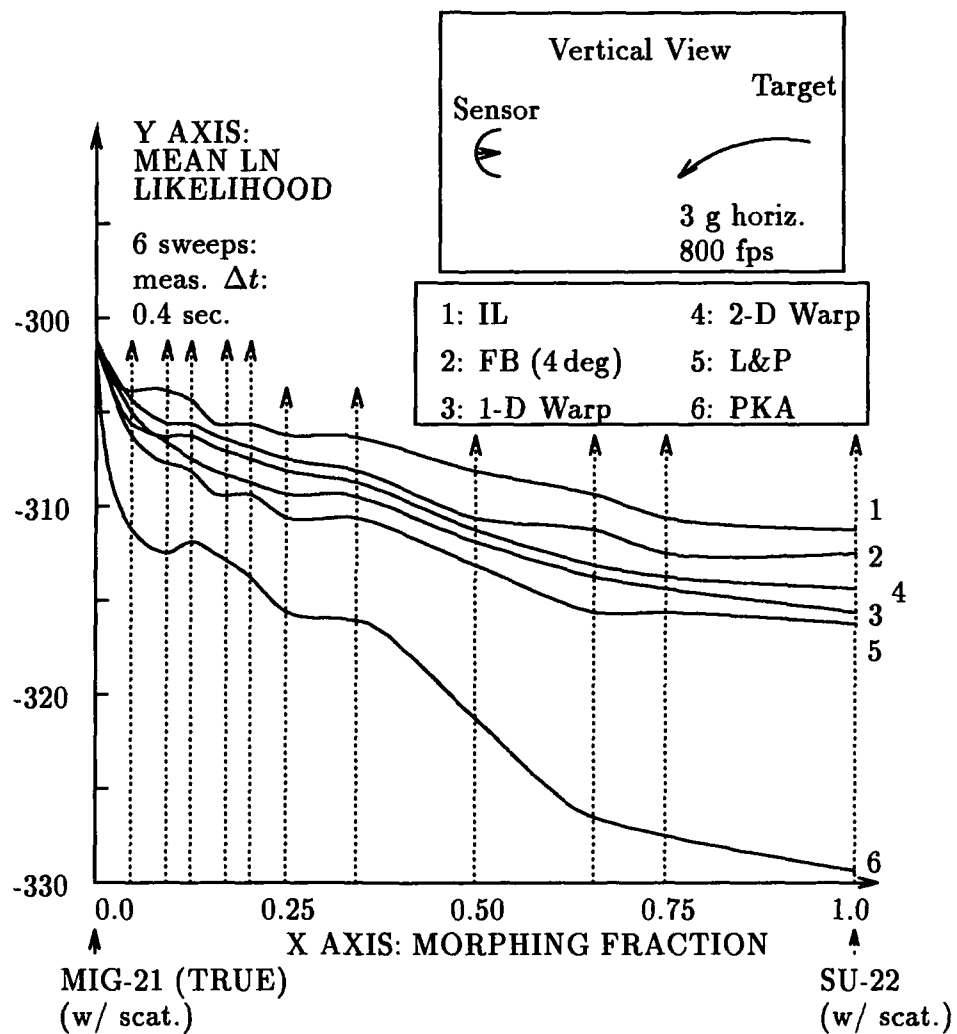


Figure 5.32. Generalized Ambiguity Function for Case 4. Mahalanobis metric, 9 dBsm std. dev. noise.

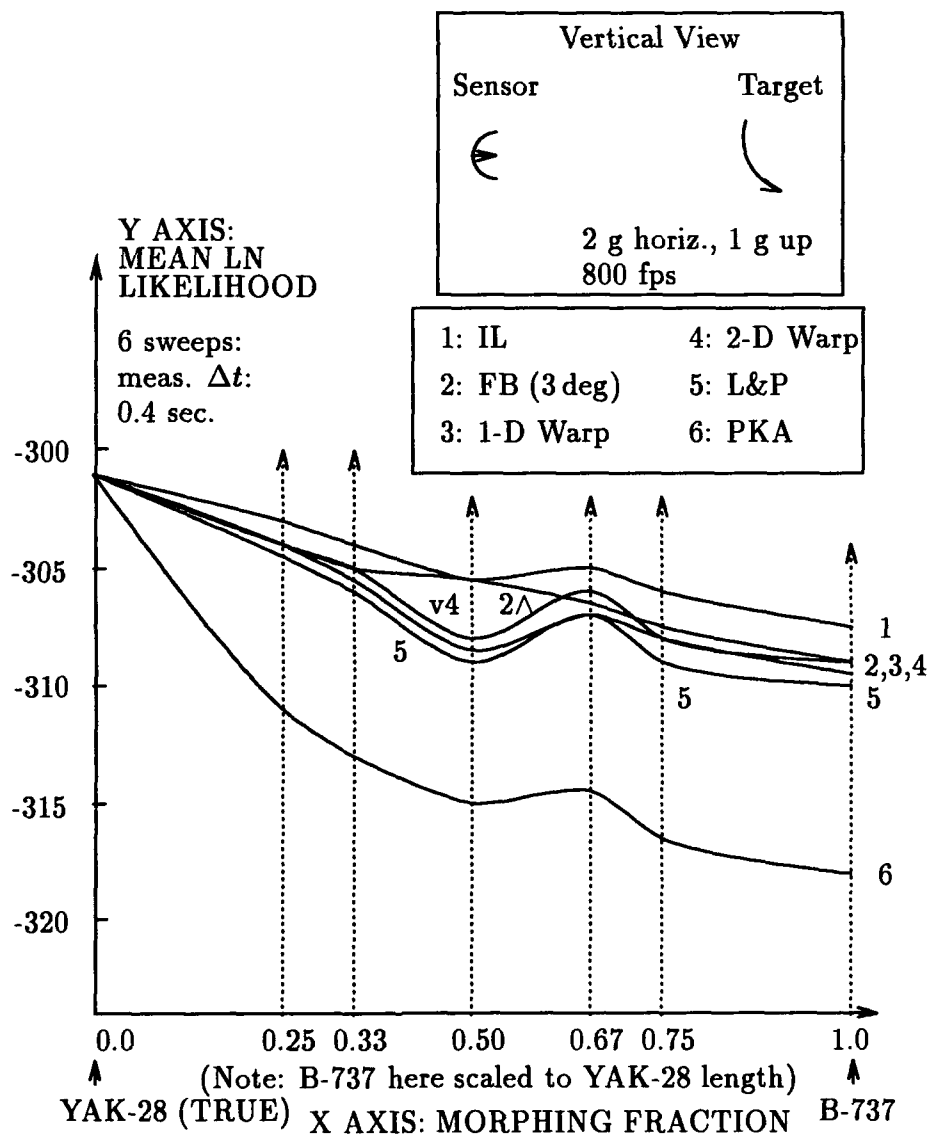


Figure 5.33. Generalized Ambiguity Function for Case 5. Mahalanobis metric, 9 dBsm std. dev. noise.

In Figures 5.34 and 5.35, we return to a case in which the origin targets are sufficiently ambiguous to cause frequent incorrect identifications with the Independent Look algorithm. In this scenario, the origin or "endpoint" targets are an F-4 with added scatterers, with and without  $4 \times 2000$  lb. bombs (two bombs under each wing), respectively. The trajectory is that of Fig. 5.33. As Fig. 5.35 graphically illustrates, this case is extremely ambiguous. Unique among cases examined in this research, these targets were so ambiguous that even perfect knowledge of origin aspect angle was insufficient to guarantee 100% correct identification, as shown in Fig. 5.34.

The classical sequence comparison or dynamic time warping-based techniques in general have not performed as well as Larson and Peschon algorithms, and their performance was particularly poor in this last, most stressing case. A detailed review of results shows clearly that this fact is largely due to a convenient artificiality in this test which real-world data may not reflect. Specifically, the measurement values in these tests are drawn from discrete locations, generally separated by two or more aspect angle "cells", or approximately 1.5 degrees. In defining an optimum low-cost path, the classical sequence comparison-based techniques used here are forced to define a contiguous path across the aspect angle space – i.e., measured signatures must in general be matched to library values perhaps considerably different from their origin locations, even on the correct model.

This factor tends to prevent the classical sequence comparison algorithms from finding as low an association cost on the *true* (unknown *a priori*) model as do other algorithms. With the likelihood value variations induced by noise, then, an incorrect model occasionally provides a higher likelihood than the true model, causing the classical algorithm to fail. This is somewhat more true in the case of the two-dimensional sequence comparison algorithm, which is allowed to find rather higher likelihood (lower cost) matches on incorrect models, while the match on the correct model tends to be the same as for the one-dimensional algorithm. The nature of the Larson and Peschon approach, on the other hand, allows it to find high likelihood (low cost) matches on the true model *and*, due to its explicit kinematic restrictions, constrains it to lower likelihood (higher cost) matches on the incorrect model.



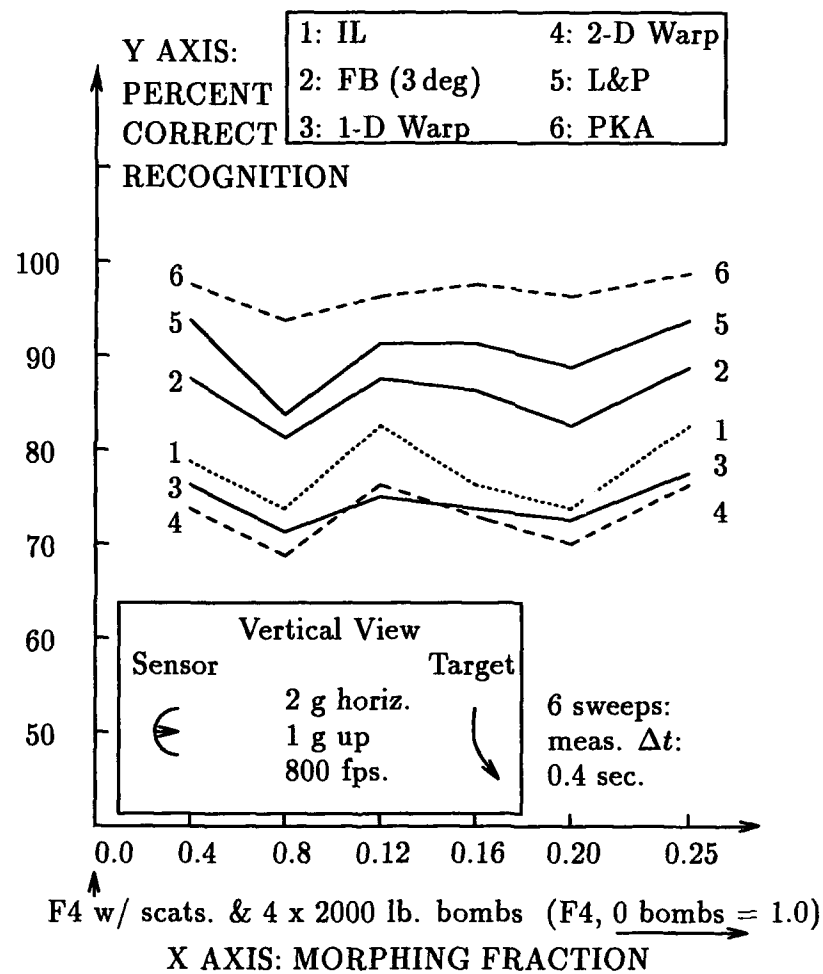


Figure 5.34. Percent Correct Recognition for Case 6. Mahalanobis metric, 9 dBsm std. dev. noise.

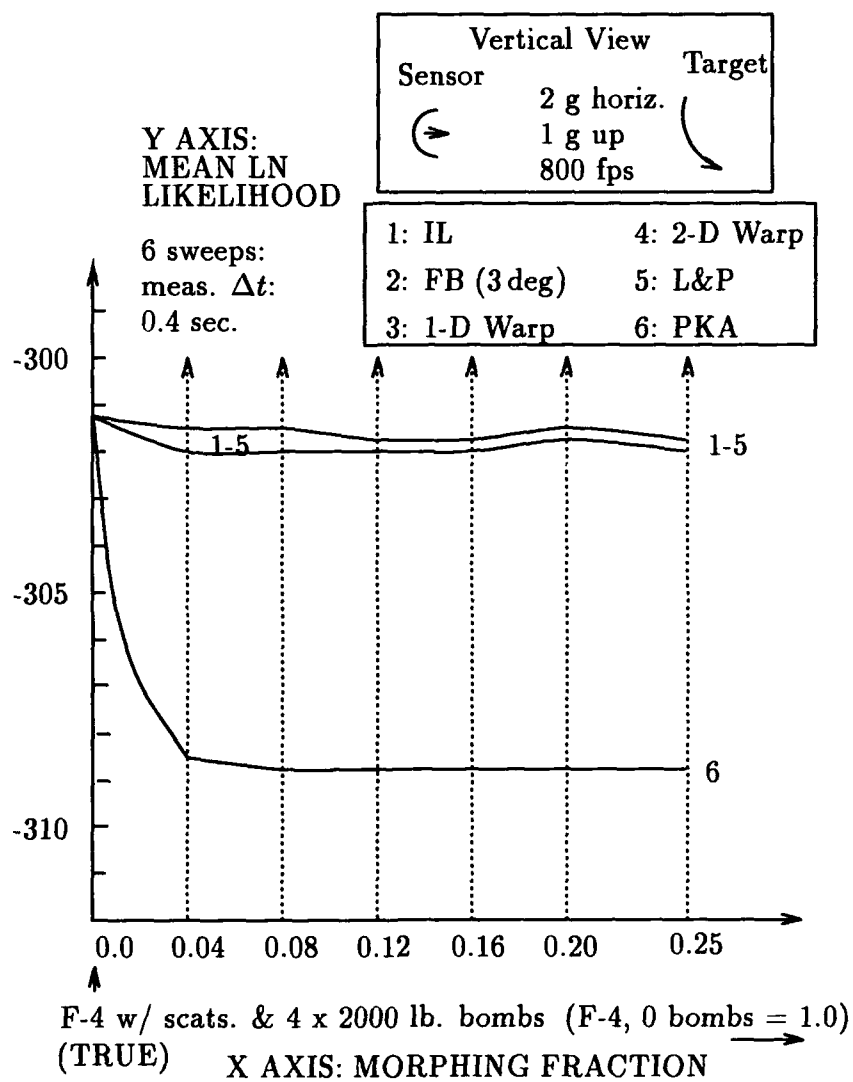


Figure 5.35. Generalized Ambiguity Function for Case 6. Mahalanobis metric, 9 dBsm std. dev. noise.

This effect can be mitigated by providing for continuity constraints which allow the classical sequence comparison algorithms to "skip" aspect angle cells with poor matching likelihoods, or by basing the output likelihood on the "best" match for each signature along the contiguous path (the latter choice is used in the previous results). Recall that, in a generic sense, the Larson and Peschon-based algorithms *are* classical sequence comparison algorithms that skip as many library cells as required to match  $n$  observations to  $n$  library cells. An essential factor to remember, however, is that allowing any dynamic programming algorithm to "skip" states brings an immediate increase in dimensionality and computational cost. Note in Fig. 3.8 that, for one-dimensional sequence comparison with the continuity constraints shown, a given library cell-measurement association has at most two predecessors – the two-dimensional case has at most six predecessors. Larson and Peschon-based algorithms are not so limited.

In any case, however, real-world signatures do not arise exclusively from one aspect angle state location, but rather are the "blurred" result of observations over small contiguous extents of aspect angle. It is expected that this blurring would tend to reduce the performance differential between Larson and Peschon-type and classical sequence comparison algorithms. Consider that dynamic time warping works in speech processing because (1) the "feature observables" extracted from speech are reasonably continuous, (2) they can be extracted with relatively little noise, (3) suitable metrics exist to distinguish differences between their realizations, and (4) the feature observable functions can be discretized or partitioned over time at intervals which preserve "closeness" in the chosen metric sense between adjacent elements in the partitioned functions. The ease with which classical sequence comparison can be employed will depend upon these factors as well, and more.

We must keep in mind that the nature and behavior of feature observables in multi-sensor fusion vary widely with each sensor and its phenomenology. Relative performance of any one algorithm in different feature spaces is difficult to predict. We may, for example, encounter a signature or feature space situation like that hypothesized in Sect. 3.6.6, in which Larson and Peschon algorithms skip highly unlikely aspect angle state locations to find a low cost match on a target model that is generally unlikely to have produced the

observed measurements. In that scenario, a Larson and Peschon-type algorithm might perform poorly.

In at least one test set using the slide distance metric, it appears that an effect of this type was observed – basically, for the scenario of Fig. 5.32, the test was repeated using the slide distance metric and noise of 5 dBsm standard deviation. The Independent Look and Fixed Bound algorithms exhibited 50% correct recognition for all morphs, or no better than a coin toss, while the 1-D and 2-D Warp algorithms exhibited 60% correct recognition or better consistently for the 67% and 75% morphs. Since it explicitly includes transition likelihoods, the Larson and Peschon algorithm was not used in this test *per se*, but a version of the Fixed Bound algorithm “biased” in the direction of motion did no better than the standard Fixed Bound algorithm. It appears that the ability to skip cells gave even Larson and Peschon-*type* dynamic programming algorithms the ability to find very low cost matches for this metric consistently on the wrong targets, while the 1-D and 2-D Warp algorithms found somewhat higher costs. For reasons noted later in this chapter, however, this observation should not be taken as representative of the performance of the slide distance metric in general.

The only statement that can be made with certainty is that restricting signature associations according to *a priori* information (kinematic or otherwise), if done properly (so as not to preclude associations required by likely behavior of the truth model), should improve recognizer performance, where incorrect recognitions are caused by unlikely associations on the wrong target model. Again, the expected order of improving performance corresponds to the order of increasing restriction for observed kinematics: IL, FB, 2-D Warp, 1-D Warp, L&P, and PKA. The next section illustrates how these algorithms and their kinematic restrictions lead to lower likelihoods for incorrect associations.

**5.6.4 Effects of Considering Likely Kinematics: Restricting Transitions.** In Sect. 2.6.2, we proposed that restricting target recognition algorithms to kinematically-feasible subsets of a matching function domain would be expected to decrease ambiguity. Fig. 5.36 shows *why* progressive domain restriction provides better separation when measurements from one target class are matched to the library for another (i.e., *wrong*) class.

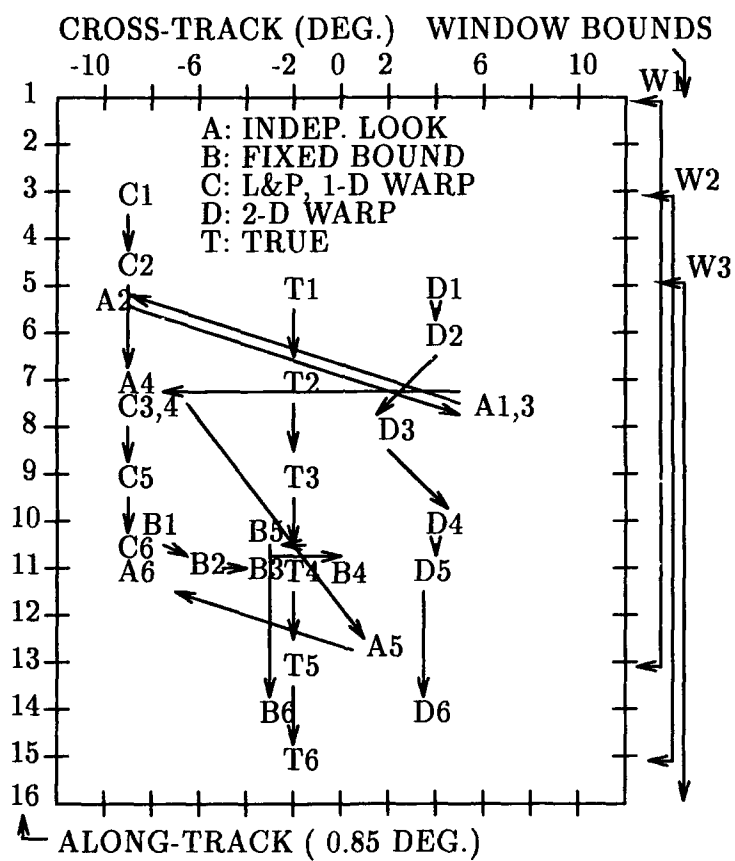


Figure 5.36. ML Aspect Angle Estimates on the Wrong Target

This figure represents a region of solid angle in target aspect defined by the union of six "windows" or aspect angle bounds for any one measurement. The "brackets" along the right side of this diagram show how the windows for successive measurement events may in general overlap (window two, or "W2", for example, extends along-track from aspect angle cell row 3 to row 15). The maximum likelihood aspect angle associations, or *pose estimates* identified by several algorithms over this angular extent for six measurements are shown – the true aspect angle is shown as well.

Note the erratic aspect sequence selected by the IL processor. Recall that the IL processor is allowed to find the best signature-to-library match (i.e., pose estimate) within the appropriate window – regardless of past (or future) associations. This erratic sequence therefore gives a lower matching cost (higher signature likelihood) than those found by algorithms which *restrict* transitions according to likely or observed kinematics. Next, note the still rather unlikely sequence selected by the FB algorithm. This algorithm requires successive pose estimates to lie within the bound limit, but does not explicitly penalize the matching cost for transitions that conflict with *a priori* expectations.

The DTW and L&P-based algorithms select more likely (linear) aspect angle paths, and their predilection to follow kinematically-reasonable paths forces a higher matching cost (lower likelihood) than given by IL or even FB algorithms for this *incorrect* model-to-target association. In contrast, when measurements were matched to their *true* target class of origin, the different algorithms are much more likely to associate with the same (true) aspect angle region, although the IL algorithm may still give unlikely pose estimate transitions, as we will shortly see.

Closing our discussion of Fig. 5.36, consider the effect of using a pose estimate extracted from this figure to drive the dynamics model in a kinematic/aspect filter of the kind described in Chapter IV. If some signature sequence-matching algorithm was able to provide a (zero error) pose estimate corresponding to points T1 through T6, then the kinematic/aspect filter associated with this target model would "fly" properly, exhibiting reasonable residuals, and so on. On the other hand, what will happen if the filter is provided with the "Larson and Peschon" pose estimate? Now it happens that, for the trajectory shown here (that of Fig. 5.27 *et al.*), error in the along-track pose estimate

corresponds closely to error in angle of attack, and error in the cross-track pose estimate corresponds closely to error in roll angle. Thus, we would expect on the order of two to 3.4 degrees error in the angle of attack estimate, and seven degrees error in the roll angle estimate. This error would contribute to "conflict" between states in the kinematic/aspect filter, which would be revealed by higher residual error for the incorrect model than for the correct model (although, for *this* case, perhaps not as large a difference as those shown in Chapter IV).

The reader may observe that use of the pose estimate sequence from the Independent Look processor in Fig. 5.36 could have an even more severe effect on the operation of a kinematic/aspect filter, even though this sequence is perhaps closer *in the mean* to the true aspect angle sequence. However, even on the correct (unknown *a priori*) target model, the Independent Look pose estimate sequence can show wild changes. This behavior is observed in Fig. 5.37, showing results from the first run made for Fig. 5.29 – a high noise, high ambiguity case. Arrows to show transitions for the dynamic programming algorithms are not shown, since these associations are all basically in the same area. Note, however, that the pose estimate sequence for the Fixed Bound algorithm still does not conform very well to the true sequence – the Larson and Peschon algorithm follows it exactly, and the classical sequence comparison algorithms differ only in minor ways. Clearly, it is better to extract this pose estimate after "smoothing" by sequence comparison methods. We will return to this subject in the following chapter.

In general, the improvement from kinematic information fusion increases with the mean aspect angle rate or  $g$  level of the target's turn. As turn rate increases, physics limits the number of possible aspect angle states  $\mathbf{x}^a$  (and therefore state sequences  $\mathcal{X}_{pk}^a$ ), and we can limit the remaining matching domain even more severely to (fewer) sequences of expected length and direction. For the FB algorithm (with a fixed sampling rate), however, we must open the aspect angle bounds to give it any chance of tracking the nominal aspect rate on the true target. This increases dimensionality and gives it a greater chance of finding an improperly high likelihood match on an incorrect target model. Other approaches for identifying infeasible aspect angle sequences may mitigate this problem, but may not effectively use the information available in observed kinematics.

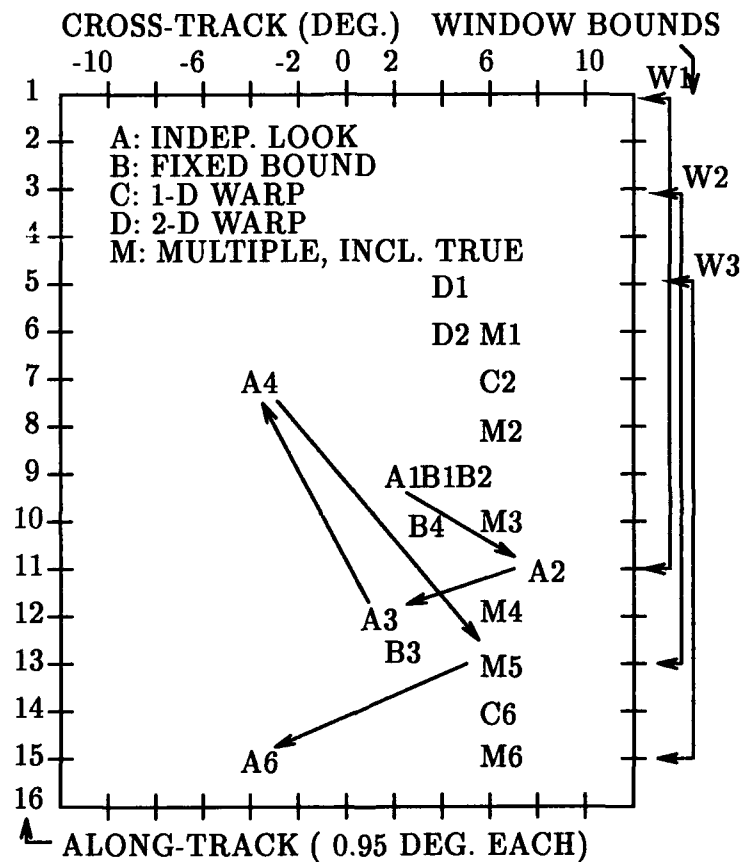


Figure 5.37. ML Aspect Angle Estimates on the Correct Target. Note: The Larson and Peschon path selected the true location in each case. Selected points for other paths, if not shown explicitly, also fall on the "multiple" point locations



Conversely, as turn rate decreases, the small mean aspect angle rate available to motion fusion algorithms tends to produce the same results as the FB algorithm, which assumes no mean rate, and can use small bounds when a small mean rate exists. For a zero-mean turn rate estimate, FB algorithms provide an effective approach – this is simply the limiting case of the L&P algorithm for a zero-mean, uniform  $p(\mathbf{x}_{k+1,n}^a | \mathbf{x}_{k,n}^a, \mathbf{Z}_m^d, \omega_i)$ . For an aspect angle rate *known* to be zero, conventional decision theoretic recognition is most effective – this is in turn the limiting case of the FB algorithm for a bound of zero degrees. This does *not* imply, however, that the decision theoretic classifier should be able to pick best successive looks from different aspect angles on any one target model – rather that the “best” *maximum likelihood* estimate of class *and* pose will be provided by the *single* aspect angle cell on one target model that yields the highest joint likelihood of originating the observed signature sequence. As motivated in Sects. 3.5 and 3.6.6, however, making a *class membership* decision using this maximum likelihood value for *one* aspect angle may be suboptimal compared to a maximum *a posteriori* estimate considering all *a priori* likely (but constant over the measurement period) aspect angle cells for each class.

**5.6.5 Miscellaneous Issues.** Effects from discretization in the aspect angle space were noticed in these tests when measurement signatures were not drawn from the library cell locations, but rather from locations in between library cell values. Although signatures preserve significant characteristics over small aspect angle changes, the Mahalanobis metric used in these tests was sensitive even to small signature changes. The primary result of this effect was a reduction in the effectiveness of sequence comparison algorithms in general and the 1-D classical algorithm in particular, since these were restricted from finding an adequately high likelihood match on the *true* target model – even on the correct model, poor discretization will drive the the Independent Look algorithm to look far afield for a minimum cost match. Tuning the sequence matching algorithms to loosen transition restrictions improved their ability to find low cost matches on the true model, but, as discussed in Sect. 3.7.2, this reduces the effectiveness of kinematic restrictions for incorrect matches. In any case, the Larson and Peschon-type algorithm retained clear superiority over the others. We should expect reduction of this discretization effect with finer dis-

cretization, a less sensitive metric, or possibly, a different feature observable space (i.e., a space which exhibits less variation as a function of aspect angle).

In addition to the Mahalanobis metric, the General Dynamics slide distance metric was also applied in these scenarios, but was found to be ineffective *in this simulated feature space* for showing differences between the various algorithms. With few exceptions (one of which was noted above) these algorithms seemed to do equally well or equally poorly with the slide distance metric. This is presumably a function of tuning, since the GD algorithm is signature-library specific, as discussed in Chapter II, and the slide distance algorithm was not *tuned* for these artificial signatures. Note that the tests discussed Sect. 5.4 were conducted under entirely different rules from those discussed in this section – longer sequences, more differences between targets, no noise, and so on. In any case, it should be clear that properly limiting the domain of any signature matching or likelihood function in accordance with observed kinematics, using any signature-to-signature metric, should not degrade correct recognition events, and may reduce incorrect recognitions significantly.

Finally, the reader should consider the value of analyzing maximum likelihood target recognition systems with generalized ambiguity functions – allowing evaluation of the “curvature” of the likelihood function around its design point. While separation at targets of interest (i.e., points of interest in parameter space) is the key design criterion, high curvature around the design point of each likelihood function should be of high secondary interest. Recall from Sect. 2.7 that this curvature is directly related to the Cramér-Rao lower bound (CRLB) for the estimator used to develop the GAF [154]: practical evaluation of this bound using our approach requires one to generate target “morphs” or interpolations arbitrarily close to the design parameter point  $\Omega_t$ , and evaluate the behavior of the GAF in this region.

As implied by the behavior of the generalized ambiguity functions in the figures shown (particularly Fig. 5.31), the limiting value of the CRLB for these estimators is evidently given by the CRLB to be found in this fashion for the PKA algorithm (i.e., joint maximum likelihood for *known* aspect angle over time). In any case, the figures make it clear that the separability of any two target classes depends on much more than behavior of the GAF around the true target parameter point. Thus, the concept of a CRLB in this construct

is perhaps not of greatest interest where we simply wish to identify a set of measurements as belonging to one of several *a priori* known points in some parameter space. Rather, the CRLB may be most useful where we wish to perform classical parameter estimation: for example, quantifying the extent to which we are able to learn the optimum location in some finite-dimensional, model-based target parameter space to represent a previously unclassified real target, known only by tracking data and signature sequences. We will return to this subject in Chapter VI.

### 5.7 Summary

In conclusion, the results of this section show that “motion warping” – the application of dynamic programming sequence comparison in moving object recognition – is a highly effective concept for multisensor fusion. This approach exploits the joint likelihood of signatures, conditioned on observed kinematics, to reduce ambiguity in recognition.

The “full” or optimal Larson and Peschon and classical sequence comparison approaches demonstrated here represent significant new additions to the previous research of Le Chevalier [136] and Mieras [164, 165]. This research shows that all of these algorithms are simply members in a large family of forward dynamic programming-based sequence comparison techniques. The final answer as to which approach is better is expected to be a function of the particular application. Classical performance tests, generalized ambiguity function analysis and consideration of computational requirements can be used to select the best technique for a particular object class, sensor, and so on.

The results given here indicate that applying the full Larson and Peschon algorithm provides more accurate, less ambiguous recognition than either suboptimal “fixed bound” Larson and Peschon-type techniques (similar to those of Le Chevalier and Mieras) or classical sequence comparison approaches. Under demanding, ambiguous target signature conditions, the Larson and Peschon algorithm best exploits “*a priori*” information on target motion. This algorithm demonstrated the ability to make correct target recognitions nearly 100% of the time, when correct recognition by an independent look processor fell to 80% or less. Suboptimal Larson and Peschon-type techniques and classical sequence comparison

approaches generally provided performance intermediate between that of "independent look" and optimal Larson and Peschon methods.

Moreover, the results of this chapter have demonstrated a new approach for establishing bounds on the performance of recognition algorithms. Treating dynamic object recognition as a problem in parameter estimation, and conducting performance analysis with generalized ambiguity functions allows definition of Cramér-Rao lower bounds for the covariance of "parameter estimates" by the recognizer.

In sum, these results provide significant new directions for the development and analysis of dynamic object and tactical target recognition. They can be used alone or in combination with other methods proposed in this dissertation to exploit the *joint likelihood* or *syntax* of observable events. The following chapter will discuss directions in which further development might proceed.

## VI. *Further Developments Exploiting Joint Likelihood in Object Recognition*

### 6.1 *Introduction*

The author believes that this work and prior contributions by Therrien [211], Le Chevalier *et al.* [136, 135] and Mieras *et al.* [164, 165] have only *begun* to assess the potential inherent in *syntactic* approaches to dynamic object recognition and other multisensor fusion applications – in particular, tactical target recognition. The author has shown that syntactic methods have great potential for identifying those target parameter sets with highest *joint likelihood* of generating observed kinematic *and* sensor signature events. The purpose of this chapter is to discuss particular directions and tasks for extensions to this research.

The two primary directions of this research – Steps One and Two as proposed in the introduction to Chapter III – were the use of (1) conventional multiple model parameter estimators with kinematic/aspect-angle trackers to assess the joint likelihood of observed kinematics, conditioned on feature observable measurements, and (2) dynamic programming-based sequence comparison techniques to assess the joint likelihood of measured feature observables, conditioned on kinematic measurements. The most important extension required to this research is to investigate Step Three – a new estimator structure which combines the two. This is the principal topic of the first section below.

### 6.2 *Sequence Comparison Methods for Single Object Identification*

In this section, we concern ourselves with the problem of tracking and classifying a single dynamic object, i.e., the case in which all measurements are unambiguously associated with one origin object. As discussed above and in Sect. 3.9, great promise exists for object recognition algorithms combining (1) classical sequence comparison and/or Larson and Peschon approaches for sequential signature processing with (2) a Kendrick-type kinematic/aspect-angle tracker or even a standard kinematic tracker. Several possibilities are suggested in the following subsections.

**6.2.1 *A New Class of Estimator for Object Recognition and Tracking.*** Recall from Sect. 3.6 that optimal use of the Larson and Peschon equations requires explicit knowledge

of the *a priori* likelihood of transitions on the aspect angle space, or  $p(\mathbf{x}_{k,n}^a \mid \mathbf{x}_{k-1,n}^a, \mathbf{Z}_m^d, \omega_i)$ . For aircraft targets, this information requires reasonable estimates of the target acceleration states. Conventional kinematic trackers cannot provide this information in real time, and their outputs require some form of smoothing to derive reasonable aspect angle estimates. Kinematic/aspect-angle trackers, on the other hand, *can* provide reasonable acceleration estimates in real time.

Conversely, the kinematic/aspect tracker requires target pose estimates, which yield the aspect angle pseudo-measurement. A Larson and Peschon-type algorithm provides this pose estimate as  $\hat{\mathbf{x}}_{k/\omega_i}^{LP}$ , i.e., the latest or  $k$ -th aspect angle state in the Larson and Peschon (LP)-derived maximum likelihood sequence of aspect angle states  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$  for object class  $\omega_i$ .

The evident next step, then, is to link a Larson and Peschon-type estimator and a Kendrick/Maybeck/Reid-type estimator, so that each provides the information required by the other. The result is believed to constitute a new form of estimator. This estimator allows one to process (1) information which does not conform to rules appropriate for linear or quasi-linear estimators, like radar signatures from an aspect angle space, in conjunction with (2) information that does conform to those rules, like range and pointing angle measurements from an object moving in physical space. An equivalent, completely linear or quasi-linear estimator may be impossible, and a possibly feasible Larson and Peschon structure to process both kinematic and feature measurements would have very high dimensionality indeed. The proposed Larson and Peschon / linear filter structure would seem to be a feasible and efficient way to process all available information.

Time constraints in this research, and the disjoint structure of simulations built for Steps (1) and (2) did not permit demonstration of the estimator approach for Step (3). When demonstrated, the resulting algorithm will allow one to combine explicitly (1) the likelihoods of observed kinematics and kinematics-related properties (e.g., mass), conditioned on observed signatures, with (2) the likelihoods of observed signatures, conditioned on observed kinematics, for each candidate object class. This will be a true *joint likelihood* object recognition algorithm using all available information, and conforming to well-understood classical parameter estimation practice.

Through this approach, linear or quasi-linear assumptions and requirements common to conventional parameter estimation could be relaxed considerably where necessary. Inherently, however, like conventional parameter estimation approaches, a recognizer of this form will consist of a set of estimators, one for each expected or otherwise desired parameter set, combined in a Bayesian multiple model estimator structure – a multiple model *adaptive* estimator structure, if the estimators are designed to alter their parameters according to observed results.

#### 6.2.2 *Other Approaches for Object Recognition with Sensor Signature Sequences.*

The primary thrusts of this research – sequence comparison with dynamic programming, residual analysis using kinematic/aspect state estimators, and the combination of the two proposed in the previous subsection, are by no means the only approach for combining this information. A completely different approach, for example, would be to treat the pose estimate history (Sect. 2.2.1) and the kinematically-estimated aspect angle history (Sect. 2.3.3.1) as separate tracks on the two-dimensional surfaces of the hypothetical aspect angle spheres for each of several candidate objects. Then, classical track-to-track association schemes [33, 10, 218] could be used to determine which track-to-track association was best, implying a correct choice of object class. For objects and feature spaces where pose estimates are reasonably well-behaved (which is generally *not* the case with HRR radar), and perhaps in other cases, this approach might be workable, and have low computation requirements.

6.2.3 *Sequence Comparison for A Priori-Likely Sequences.* The use of dynamic time warping for word detection in continuous speech [176] suggests a method for detecting aircraft target roll maneuvers at the time of their occurrence, i.e., prior to the development of normal load acceleration and kinematically-observable events. Basically, if we are tracking an aircraft, and have a reasonable idea of its class and current aspect angle, we can construct sequences of likely sensor signatures corresponding to particular roll maneuvers. These sequences will then be continuously compared with the observed signatures – the onset of a roll maneuver will be marked by a sudden “match” between the incoming signature sequence and one of the constructed sequences. In effect, this is a form of multiple

model estimator. Recognition of such a roll event would cue the kinematic tracker for changing conditions.

A development of the "fixed bound" Larson and Peschon or Le Chevalier-type syntactic algorithms [136] (i.e., requiring no *a priori* information other than bounds) would also find use in this application. Applying such an algorithm to sequences of observations from the target would, for a stationary aspect angle, indicate a sequence of maximum likelihood associations in approximately the same aspect angle area. In a turn event, the sequence of associations would proceed off in a direction indicative of a turn – the change in associated aspect angle would be derived readily from the algorithm. The path direction should not be expected to be as accurate as would be the case if *a priori* information were available from kinematics, but it may be accurate enough to cue the kinematic tracker.

Extensions of this concept would use dynamic programming-based methods or other sequence comparison techniques for identifying *any* object for which particular feature sequences can be predicted or expected, in the absence of an ability to estimate the underlying kinematics or other state transitions directly. Some tactical targets have "canned", operator-learned, or otherwise *a priori*-likely maneuvers which may generate characteristic feature sequences under circumstances in which target kinematic state measurements are not available, or in which *observable* kinematic states are not highly correlated to the feature values. For example, particular exoatmospheric or re-entering objects have characteristic signatures imparted by periodic or reasonably predictable behavior. Other speech processing-related techniques have already been considered in this particular strategic defense-related area, such as Therrien's use of Linear Predictive Coding (a technique often used in speech analysis) to model target signatures [211].

### 6.3 Sequence Comparison Methods for Multi-Object Tracking and Data Association

Here we consider the problems of tracking and classifying multiple objects, i.e., the case in which each measurement *cannot* be unambiguously associated with any one origin object. All of the techniques and algorithms discussed in this research for exploiting the joint likelihood of observed events and known object classes are applicable to these problems. Some cases of particular interest are given below.



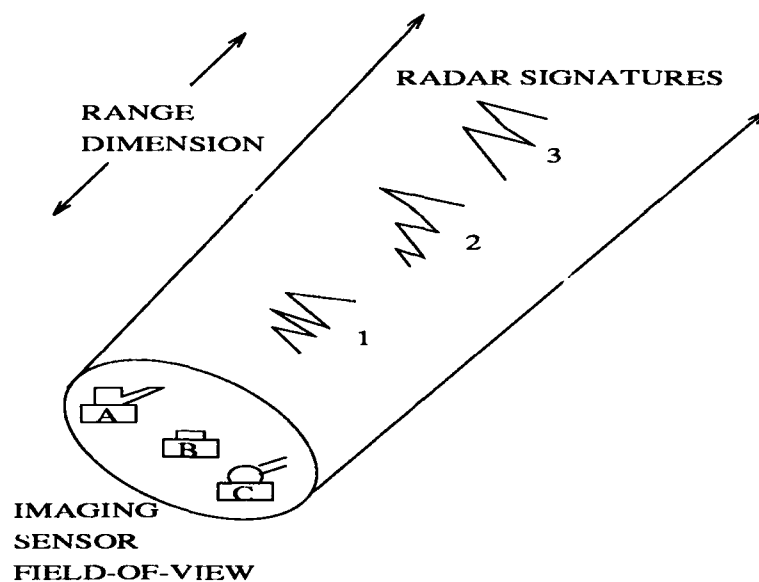


Figure 6.1. Problem: Associating Signatures From Imaging and Radar Sensors (figure inspired by [96])

**6.3.1 Motion Warping for Observation-to-Observation Association.** Classically, the spatial resolution capabilities of radar and passive optical sensors are counterposed – that is, radar sensors have limited angular resolution, but excellent range resolution, while passive optical sensors have excellent angular resolution, but no range resolution (excepting secondary methods like stadiametric range finding, based on knowledge of object class, orientation, and angular extent in the image plane, etc.) At least one recent article [96] notes that the problems of associating objects between imaging and radar sensors are not completely solved.

Consider the situation described in Fig. 6.1, taken essentially from the last reference. This situation could arise, for example, in observing three moving objects on a relatively flat surface at long distance from a remotely-piloted vehicle equipped with low-cost imaging and radar sensors. We wish to direct artillery fire on object (A), but are unable to determine its range unambiguously, since the angular resolution cell of the radar encompasses the three objects in the imaging sensor field of view, and arbitrary orientation of the objects makes stadiametric range finding impossible.

Using the approach applied in Chapter V, however, we can say that a total of three possibilities exist for associating object (A) with the range signatures. If the object class of (A) is unknown, then at least in general we can expect that (A) is a member of one class  $\omega_j$  of  $J$  *a priori* likely object classes. Each of these possibilities defines a particular object-trajectory combination on the ground surface (which may be more or less well known), and therefore implies a particular sequence of signatures in the image and radar domains. The maximum likelihood association of (A) with (1), (2), or (3) and object class  $\omega_j$  of  $J$  classes should be the one for which the joint likelihood of observed kinematics and signature sequences (in image *and* radar domains) is greatest. Thus, the approaches defined in this research may provide solutions for problems now considered to be unsolvable.

An extension to this fusion process might provide for feedback to improve the sensor outputs themselves – for example, by providing a “best” choice for a model and time-varying orientation to be used in a model-based segmentation algorithm [27, 29].

**6.3.2 Joint Likelihood in Observation-to-Track Assignment.** As in the preceding problem, one could augment the decision process for observation-to-track assignment with “motion warping” and related means for exploiting the joint likelihood of kinematic measurements and sensor signatures. This development should provide a tangible improvement over state-of-the-art approaches as discussed in Sect. 2.3.3. Inherently, the maintenance of a track history gives a departure point for likely kinematics, which imply likely feature observations, and so on. At least, this research should persuade researchers to treat “nonkinematic” variables and measurements as interdependent with “kinematic” ones, and provide options for exploring their relationships in multisensor fusion.

#### **6.4 Mathematical Issues in Multisensor Fusion for Recognition**

In an early article on pattern recognition, Ho and Agrawala [108] observed that many concepts well understood in the context of state and parameter estimation were applicable in pattern recognition. This remains true, and in this section we will discuss particular directions that evidently remain to be explored regarding the application of state and parameter estimation concepts to dynamic object recognition.

*6.4.1 Parameter Estimation and Object Recognition.* In general, current object recognition efforts tend to focus on assigning sets of measurements from an unknown object to one element in a finite set of *a priori* known object classes. It is believed, however, that a potentially fruitful area of research would be to consider *continuous* object parameter spaces, and the use of measurements to define points in these parameter spaces with maximum likelihood of having generated the measurements. As discussed in Sect. 3.11, real object parameter spaces are infinite-dimensional, but we are forced to model them in finite-dimensional spaces, choosing levels of model fidelity that match real objects to within some distance in some norm or distance measure. Using the mathematics of functional analysis [170], it should be possible to pose and answer many elegant questions regarding our ability to model or identify real objects using finite-dimensional models.

The practical application of this research is that it allows us to consider object identification when the unknown object class is *not* one of those well characterized *a priori*. We may obtain measurements of a completely new object, and be able to make conjectures about its parameter set in our finite-dimensional model space, based upon well-characterized points in the parameter space which correspond to known objects. This is no more than a proposal to use well-known parameter estimation concepts [154] for estimating or learning object parameters during recognition.

As noted in Sect. 2.7, the ability to compute an ambiguity function gives us the ability to define a Cramér-Rao lower bound on the parameter estimate error covariance matrix. For a set of parameters "learned" as proposed in the previous paragraph, this lower bound gives us a measure of confidence regarding the quality of our estimate. Even where we do not desire to learn parameters, this covariance bound should give us, for two arbitrary loci of points in state/parameter space corresponding to well-characterized object classes engaged in some particular maneuver, an understanding of how well we can expect to discriminate between them with our classifier. Due to the limited number of observations likely in a tactical scenario, this will require not only a well-shaped ambiguity function, but also likelihood functions which have probability density functions closely held around the mean, or ambiguity function, value.

*6.4.2 The Motion Warping Process as a Nonlinear Functional.* As discussed in App. B, a nonparametric linear classifier is a *linear functional* [170] – that is, a linear function, operating on a vector (feature) space over a scalar field (the real numbers, in this case), the output of which is an element from the scalar field (a distance). The mathematical structure of linear functionals prescribes the existence of a vector space *dual* to the feature space, in terms of which dual space the parameters of the functional can be stated – specifically, the equation of the hyperplane corresponding to the linear classifier is described by a scalar and a point in the dual space, which is equivalent to the first  $n$  dimensions in the  $n + 1$ -dimensional space called the *weight space* by Tou and Gonzalez [212].

Similarly, the motion warping processes and other joint likelihood methods described in this research are nonlinear functionals – nonlinear functions, operating on a Cartesian product space of aspect angles, features, and kinematic measurements, to produce a scalar output (likelihood, or perhaps simply warping path distance). Many of these algorithms are subject to further classification as functionals: for example, for the feature observable-to-feature observable (e.g., range sweep to range sweep) Mahalanobis metric employed in this research (which treats the range sweeps as elements in a finite-dimensional Hilbert space), it can readily be shown that motion warping is a *bounded* nonlinear functional [170:344]. These observations may provide a path for further theoretical analysis of the concepts proposed in this research.

*6.4.3 Sequence Comparison for Multiresolution Analysis.* Recent efforts have been directed toward the use of wavelets, optimal estimators, and other tools for analysis of multiresolution or multiscale processes [150, 17, 54]. In this regard, it should be noted that the sequence comparison techniques discussed in this dissertation inherently have potential for multiresolution applications. For example, one can conduct sequence comparison processes for a range of “fineness” or spatial discretization levels. The results of these comparisons may reveal the presence of grammatical or syntactic structures ranging from coarse to fine in the space over which the comparisons are conducted. Iterative algorithms may provide for progressively finer investigation of the likelihood that particu-

lar spaces generated particular sequences or subsequences, ceasing investigation when finer resolution appears pointless.

### *6.5 Alternative Approaches for Sequence Comparison.*

The research detailed here has focused on forward dynamic programming-based techniques for sequence comparison, but these are not the only tools for this task – there are many methods for comparing sequences. The principal competitor to dynamic time warping in the speech recognition problem is the hidden Markov model (HMM). In turn, the HMM can be considered as a form of stochastic automaton [176:309], which, like dynamic programming-based sequence comparison, is a key tool in general syntactic pattern recognition, as discussed in Chapter II. Another alternative class of approaches is offered by neural networks. The author chose not to investigate these alternatives, primarily due to what he perceives as disadvantages to them, as alluded to in Chapter I. Subsequent to the start of this research, some aspects of these approaches have been investigated at AFIT [128, 68, 83], as noted in Chapter II, and independent effort using these methods has been undertaken elsewhere [81, 199].

A key issue to be addressed by researchers applying HMM's, neural nets, and other "trained" algorithms to this problem is how to train the algorithms for a variety of expected scenarios without introducing ambiguity into the recognition process. There may be a need for research to define the tradeoffs between maintaining large libraries from which sequences can be constructed at will (as in this research), versus maintaining large numbers of HMM's and/or neural nets, each defined for a particular set of events.

### *6.6 Miscellaneous Issues and Extensions To This Research*

Aside from the further algorithm developments proposed above, a number of perhaps more mundane but critical remaining issues can be explored using the basic algorithms employed for this research. These include, but are not limited to, the following:

*6.6.1 Feature Space and Distance Metric Choices.* This research has focused on the use of "motion warping" or dynamic programming-based sequence comparison for

moving object recognition using a feature space of high range resolution radar sweeps. The method, however, should be applicable to any of the global descriptor features, as generated by imaging or other sensors, discussed in App. B. A particularly powerful approach would appear to be that of making syntactic or motion warping object recognition based on simultaneous analysis of feature observables from *multiple* domains – infrared, visual light, and ultraviolet spectrum or image-extracted features, narrowband and wideband (high-range resolution) radar, and so on. The objective here would be to exploit the concept of joint likelihood further by making use of the processes in two or more feature domains (ideally, stochastically independent ones) to reduce the likelihood further that an incorrect object class could generate observed feature sequences.

*6.6.2 Application to Other Classes of Objects.* The discussion in this dissertation has been primarily oriented toward air and ground tactical targets. The principles and analysis tools discussed here, however, should be applicable to many other object and target classes – animate beings, ships, satellites, etc. The requirement is simply to have some set of restrictions which couple state transitions and measurement generation according to known rules (parameters unique to each class), combined with measurements from multiple sources that will *conflict* when incorrect parameter sets are assumed.

Also, although we have spoken most often with regard to dynamic object recognition in this effort, it should be clear that the principles behind this research also apply to the recognition of *stationary* objects. Simply, if it is *known* that a given object is stationary in aspect angle and/or translation with respect to the sensor, one should not match noisy sensor signatures from that target with library models in a way that allows or requires the models to translate and/or rotate between signatures. To do so is to invite an incorrect recognition.

*6.6.3 Path Discretization.* In virtually every discussion of dynamic programming-based processes, the discussion recognizes that continuous spaces must be discretized, but gives no guidance as to the appropriate discretization fineness. In Sect. 5.6, we discussed one effect of inadequate discretization observed using the RCSToolLLBox-generated feature space and the Mahalanobis metric. The sensitivity of the dynamic programming-based

algorithms proposed here to discretization fineness should be further evaluated. The ambiguity function techniques discussed above provide a natural, if empirical, way to conduct such analysis.

That is, considering likelihood function values due to feature observable matching only, one attribute of a proper discretization level may be the absence of a difference at the *true* parameter point  $\Omega_t$  between ambiguity function values for sequence-matching and non-sequence-matching algorithms. This means that, on the true target, there is no cost penalty for sequence matching algorithms due to "false" pose estimate motion on the target, like that caused when the matching process "hunts" to find the best cell from the available discrete set which might have generated the observed measurements from the underlying continuous space (in our case, aspect angle).

**6.6.4 *Alternative Kinematic/Aspect-Angle Filter Designs.*** The effects and possibilities noted in Chapter IV deserve to be repeated using filter models with higher fidelity – in particular, with parameters and aspect angle pseudo-measurements matched to real objects and sensors of interest. This proposal is made with regard to a conventional kinematic/aspect-angle estimator construct, i.e., one that does not explicitly predict and compare feature observable measurements.

**6.6.5 *Application of the Larson and Peschon Equations in Speech Recognition and Other Areas.*** In Sects. 2.4.5 and 3.8, we compared the Larson and Peschon equations to elements from the family of classical dynamic programming-based sequence comparison methods. We noted that the Larson and Peschon equations are, from a general perspective, simply one member of this family, distinguished from other members by very special continuity constraints. This observation leads one to consider the possibility of applying the Larson and Peschon equations in classical sequence comparison tasks, like speech recognition. The immediate advantage to consider, as observed in the referenced sections, is that "Larson and Peschon"-type sequence comparison would not be penalized for length differences between two sequences to be compared. As we have seen in Chapter V, subject to its *a priori* guidance on sequence element separation, the Larson and Peschon approach can readily match a very short sequence of  $n$  elements to the "best"  $n$  elements from a

very long sequence of  $m$  elements ( $n \ll m$ ). From that result, we might desire in some fashion to compare the "skipped" elements of the longer sequence to the nearest elements of the shorter sequence, but at least we have a good starting point – much better than we would be likely to achieve with classical "continuous" sequence comparison.

For other possible applications of the Larson and Peschon approach (and other aspects of the research presented here) with regard to ISAR radar and related areas, the reader is referred to a previous article by Le Chevalier [137]. Note that the referenced article is classified and in French, without translation in the origin document.

### 6.7 Summary.

This chapter has attempted to point the reader toward promising areas for fundamental research, and to list applications for the approaches and algorithms developed in this effort. It is clear that much more can be done to apply syntactic pattern recognition and classical parameter estimation techniques to multisensor fusion and object recognition.

Syntactic or joint likelihood approaches to object recognition should *always* be considered when the objects of interest obey known rules that couple behavior (state dynamics) and appearance (measurable or observable quantities) in characteristic ways (as functions of parameters unique to each object class). This chapter shows promising directions for applying the results of this research and other methods toward syntactic solutions to current problems.



## VII. Summary and Conclusion

This research explored methods by which the obvious interrelationships between target motion and sensor signatures could be exploited for recognition of dynamic objects in general and tactical targets in particular. A wide variety of methods and directions were taken in this effort, but all culminate in one basic result – producing likelihood functions that exploit the *joint* or *coupled* nature of physical processes over time for known object classes, to reduce *ambiguity* in deciding the most likely object class to have produced a given sequence of measurements or observations.

This research can be understood from at least three different, yet fundamentally equivalent perspectives. First, from a probabilistic standpoint, we have defined new expressions for the *joint likelihood* of observed events from known target classes, conditioned on past measurements and *a priori* information for each class. Unlike conventional pattern recognition approaches, we are specifically interested in the joint likelihood of events *over time*, since we wish to consider the likelihood of physical processes implied by those events. Ideally, these likelihoods would be used with Bayesian methods to estimate an *a posteriori* probability of class membership for an unclassified target, conditioned on all available information.

Second, this research can be posed as an extension to the theory and practice of *syntactic* pattern recognition – that is, the class of algorithms which distinguish entities using order or structure, commonly by representing an observed entity as a sequence of elements over time, and assessing the likelihood that any known class could generate the observed sequence. For known physical objects, the process of generating sequences or sequence spaces of expected elements for each class inherently considers the *joint* or *coupled* nature of the processes which produce the sequences. We may then apply a range of available syntactic sequence comparison tools to associate the observed sequence with its most likely origin class.

Lastly, this research can be considered as having defined techniques for *restricting the domains of likelihood functions* used to identify known object classes, making restrictions according to joint or coupled aspects of behavior expected over time. Restricting

the domain of an object class-specific likelihood function or matching function tends to prevent that function from improperly associating itself with a sequence of measurements that should not, all facts considered, have come from that object class. For the correct combination of observed sequence and likelihood function, proper domain restriction will not hinder the association process.

The results of this research are major extensions to the theory and practice of sequence comparison methods in dynamic object recognition, as pioneered by Therrien [211], Le Chevalier [136] and Mieras [164, 165]. Furthermore, they strongly confirm the observations of Agrawala and Ho [108] and Daum [9:177–178] regarding the potential applicability of state and parameter estimation techniques in object recognition.

The theoretical and practical contributions of this research are restated below from Chapter III, and demonstrated results from Chapters IV and V are referenced to support each claim. They include:

(1) Extension of conventional multiple model residual sequence analysis techniques and kinematic/aspect-angle trackers to provide new methods for object and target recognition, in particular where sensor measurements are not linearly predictable. The techniques described in Sect. 3.4 and demonstrated in Chapter IV represent a completely new approach to recognition of dynamic objects in general and aircraft targets in particular. The potential effectiveness of these methods can be gauged by comparing the likelihood values for correct and incorrect associations in Figs. 4.9 and 4.10, or the state estimates for correct and incorrect associations in Figs. 4.5 and 4.6.

(2) Extension of the Larson and Peschon equations [133] (Sects. 2.4.4 and 3.6) to provide new methods for dynamic object recognition using measurements from ambiguous feature observable spaces, considering – in an optimal fashion – *a priori* information from kinematics and other sources as to the likelihood of transitions on the underlying aspect angle state space. The reduced ambiguity provided by this approach over conventional “independent look” processes is demonstrated in Figs. 5.27 through 5.35 – up to 20% improvement in percentage of correct recognition is observed in this particular application.

(3) Extensions of the theory and practice of classical dynamic programming-based sequence comparison to include feature observable sequences arising from an aspect angle subspace. The reader may compare Figs. 2.8 and 3.8 to observe this extension graphically. Results showing reduced ambiguity for these algorithms compared to conventional approaches are seen also in Figs. 5.27 through 5.35.

(4) Combination of the of the Larson and Peschon equations with conventional linear estimators to provide a *new* form of estimator, suitable in particular for object recognition with ambiguous feature observables, generated from dynamic subspaces that exhibit linear behavior in some respects. This process is described in Sects. 3.9 and 6.2.1.

(5) Through contributions (1) through (4) and application of Bayes' Rule, several *new* approaches for multisensor fusion to obtain *a posteriori* probabilities of object class membership, conditioned jointly on kinematic and "nonkinematic" or feature observable information and *a priori* information on each known object class. In particular, Eqn. (3.18), a development of the Larson and Peschon equations, is a completely new expression for *a posteriori* probability of target class, allowing the user to fuse ambiguous sensor signatures and kinematic information in an optimal fashion for object recognition. Other relevant expressions are Eqns. (3.2) and (3.21). These equations can be readily used with likelihood values generated as shown in Chapter IV and V.

(6) Identification of a *new* method for evaluating object recognition algorithms – the generalized ambiguity function. This approach provides a useful alternative to the conventional performance approach of evaluating probabilities of correct and incorrect recognition. Figs. 5.27 through 5.35 include several examples of performance evaluation by generalized ambiguity functions.

(7) Extension of classical parameter space concepts into the field of object and target recognition. In particular, the discussion of Sect. 3.11.3 and the target model generation procedures in Sect. 5.3, combined with later results in Chapter V, motivate the need to consider objects as discrete sets of parameters in continuous/discrete parameter spaces, and demonstrate new methods for allowing that consideration.

(8) Through contributions (6) and (7), identification of a *new* and practical approach for obtaining a Cramér-Rao lower bound for dynamic object and target recognition algorithms. The potential in this approach can be noted by comparing the discussion in Sects. 2.7 and 3.11.3 with the generalized ambiguity function plot of Fig. 5.31.

In sum, this research provides a wide range of *new* approaches for fusing kinematic and "nonkinematic" or sensor signature information in dynamic object recognition, and evaluating the results of that fusion. Multisensor fusion of target kinematic and sensor signature information remains an exceptionally rich and promising field. The research described here has illuminated significant new directions for research in multisensor fusion and target recognition. First, multiple model estimators and syntactic sequence comparison methods offer powerful tools for exploiting the joint likelihood of physical processes over time. Second, treating tactical targets as collections of parameters executing particular state transitions over time allows one to apply a very large family of existing estimation and control approaches to dynamic object and target recognition. The methods used in this research are but members of this family – many other methods await application.

## Appendix A. Definitions

**Algorithm Q** An algorithm developed by General Dynamics [20] for target recognition by high range resolution radar, or the physical realization of this algorithm as a set of Fortran subroutines. See *slide distance*, below (note: the elements of General Dynamics responsible for this effort were absorbed by Hughes in 1993).

**ambiguity function, generalized ambiguity function** In general radar usage, an ambiguity function [57, 15, 16] is a real function of two variables, time delay and frequency, which provides an indication of a radar's ability to distinguish between two targets with different ranges and/or doppler shifts. Ambiguity functions are used in radar pulse waveform design. The generalized ambiguity function (see Eqn. (2.37)) is the expected or mean value of a generalized likelihood function defined for various conditions, taken over all possible measurement values for some set of true conditions, and provides an indication of the ability of a likelihood function to distinguish between two state/parameter sets.

**angle of attack** The angle, generally and in this work denoted  $\alpha$ , between the body  $\vec{x}_b$  vector and the projection of an aircraft's wind vector (or velocity vector, for an atmosphere at rest with respect to the inertial frame as assumed in this work) onto the body  $\vec{x}_b - \vec{z}_b$  plane. See Fig. 5.23.

**a posteriori** The definition of this term of interest to us, from the several provided by Webster, is "based on observation or experience; empirical" (as opposed to "*a priori*"). In our case, this term refers to the whole or sum of information available about some event or system *after* additional or new information is extracted from some set of measurements or observations and combined with *a priori* knowledge.

**a priori** The definition of this term of interest here, from the several provided by Webster, is "before examination or analysis" (as opposed to "*a posteriori*"). In our case, this term refers to information available about some event or system *before* additional or new information is extracted from some set of measurements or observations. A *a priori* knowledge may be based on previous measurements or observations, theoretical knowledge, other sources, or may not exist.

**aspect angle path (true or estimated)** The path traced on the *hypothetical aspect angle sphere* (see reference below) of unit radius by the object-to-sensor unit vector. May be true, based on true object orientation and sensor location, or estimated, as from kinematic measurements, state estimation, and assumptions as to orientation required for given kinematics.

**aspect angle region, subset** See *aspect angle space* first. A region on or subset of the hypothetical aspect angle sphere (of unit radius). If an aspect angle region or subset is one-dimensional, it is equivalent in description to an aspect angle path.

**aspect angle space** The entire  $4\pi$  steradian (two-dimensional) extent of the hypothetical aspect angle sphere (equivalently, the surface of the unit sphere) is closed under allowable transitions on that sphere, and is therefore considered an aspect angle "space." A third dimension of aspect angle ("roll" about the sensor-object vector) may be required to specify angular relationships for signatures that are not in-plane rotation invariant, or where the full three degree-of-freedom rotational state of the object is under consideration. See Sect. 3.2.

**aspect angle sphere** See "hypothetical aspect angle sphere."

**bank angle** An Euler angle about the body frame  $\vec{x}_b$  axis, last in a series of three Euler angles (azimuth, elevation, and bank) required to rotate the navigation (inertial) frame into the body frame. This is not the same as *roll angle* (see *roll angle* below and [79]).

**bijective** A term used to describe a mathematical function that is both "one-to-one" and "onto" (these terms defined below).

**bin** Generally, an interval in some space, usually one interval of a set of equal-width intervals defined by a partitioning of a simply connected subset (a larger interval) of a one-dimensional space. Values of some function over the space at some point or points which fall within the interval of a given bin may be assigned, summed, or integrated into that bin, allowing the distribution of the (possibly continuous) function values to be approximated by discrete bin numbers and values. Representation of observed or measured function values as bin "return" values is often forced by limited resolution

of the measurement system (i.e., sensor resolution equivalent to bin width) or a desire to use clustering techniques on the output.

**cross-correlation** In stochastic processes, a measure of the interdependence between two (possibly vector-valued) random variables, found by taking the expected value of the outer product of the two variables over all realizations [153:93]. The one-dimensional signal (cross-) correlations  $c(n)$  for two signals or images  $a$  and  $b$  represented as discrete sequences (see [93, 99, 178]) or vectors of  $K$  or more elements are defined by equations of the basic form:

$$c(n) = \sum_{k=1}^K a(k)b(k-n)$$

In general it is desired to find the value of  $n$  which maximizes this expression. This expression can be extended to provide for two or more dimensions in  $a$  and  $b$ , and to provide for normalization by the "energy" (or an analogous measure) in  $a$  and  $b$ .

**decision theoretic (pattern recognition)** Pattern recognition systems which call for each pattern class to be represented by a point or collection of points in a multi-dimensional feature space, each dimension of which is parameterized by a continuous or discrete set of values of some measurable quantity, a feature. Where more than one point is assigned to a pattern class, each point generally represents one realization of a stochastic (i.e., "random") process which generates the feature values for that class. The process is generally stochastic in four respects, due to unknown or *a priori* unpredictable variations in (1) class parameters, (2) class state, (3) environment (in particular, as affects the object-sensor line-of-sight), or (4) sensor system.

**doppler velocity (measurement)** A quantity derived from a returned waveform (generally emitted originally by the receiving sensor - that is, a *monostatic* sensor arrangement) which indicates the relative velocity between the sensor and the observed object, based on (doppler) frequency shifts in the waveform upon reflection at the observed object. For a monostatic sensor, doppler velocity is a measurement only of the component of sensor-object relative velocity along the sensor-to-object vec-

tor, and provides no information on object velocity normal to that vector. Doppler velocity can be obtained from radars of the type considered in this research [16, 203].

**dynamic programming** A technique for finding a best or "shortest" path through a discrete, finite set of possible decisions, where that path obeys the *Principle of Optimality* (see Sect. 2.4.1).

**dynamic time warping** A technique for assessing the similarity of two finite sequences of speech features or feature vectors, as "discretized" or extracted from windowed or partitioned continuous speech, by removing differences due to expansion or contraction of one sequence relative to the other (see Sect. 2.4.2).

**dynamics** The branch of mechanics that deals with forces, masses, and their relation primarily to motion, but also sometimes to equilibrium conditions (as opposed to *statics*, which deals only with force and mass in absence of motion, and *kinematics*, which is defined below; see [144]).

**extended Kalman filter** A class of mathematical estimators derived from the (linear) Kalman filter, designed to make "near-optimal" estimates of quantities for which time-propagation and/or measurement equations are nonlinear, using truncated Taylor series and relinearizations about current values (see Sect. 2.3.1.1).

**feature** This term has basically two meanings in pattern recognition, depending on the form of pattern recognition considered. In decision theoretic or syntactic pattern recognition, a feature is some quantity or property pertaining to object classes of interest which can be measured or observed, but which may or may not provide unambiguous information as to class membership. In point or pairs-of-points correspondence-based image recognition techniques, however, a feature is some portion of the object image pertaining to object classes of interest which is believed to be meaningful to the recognition process (a *landmark*), but which may or may not be easily measurable and usable by the recognizer – the object of these correspondence systems is to find correspondences between points on the unclassified object and features on a library image (see App. B).



**feature attribute** Generally equivalent to the first meaning given for the term feature.

See *feature observables* below.

**feature observables** Equivalent to the first meaning given for the term feature, this term is preferred by the author over the term feature attribute because of the emphasis given to the fact that the sensor can observe or measure these quantities (presumably a object could have an "attribute" that is not observable).

**feature space** A vector space in which a given feature observation or measurement of the object can be expressed as a point. Thus, the dimension of the feature space is determined by the number of individual scalar elements (which may not be independent in any sense of the word) in a feature observable measurement vector. Feature spaces of dimension  $n$  are generally taken to be isomorphic [170:173] to  $R^n$ , the (Cartesian) vector space defined by a collection of  $n$  mutually orthogonal real axes sharing a common origin (see App. B).

**fire control** The process of controlling the operation of a classical gun or unguided rocket-type weapon system, i.e., a weapon system firing projectiles which cannot be controlled in flight, for which prediction of target trajectory over the projectile time of flight is the critical concern in aiming.

**flight path angle** The angle between an aircraft's velocity vector (not generally equal to the wind vector) and the local horizontal plane (see Fig. 5.23).

**global descriptor** A (generally scalar) value which is in general a function of the entire (global) extent of a object, as isolated (segmented) and observed by some sensor (see App. B).

**heading angle** The angle between local north and the (vertical) projection of an aircraft's velocity (not generally wind) vector onto the local horizontal plane (see Fig. 5.23).

**heuristic** Defined by Webster as "valuable for stimulating or conducting empirical research but unproved or incapable of proof." In discussion of optimal estimation techniques, heuristic carries the meaning "not mathematically rigorous." In pattern recognition [212:18-19], heuristic carries the added meaning "based on human

intuition and experience," so that artificial intelligence-based pattern recognition procedures are considered to fall into the heuristic category.

**high range resolution (HRR)** A term used to refer to radar systems which can isolate reflected returns from one pulse into partitioned intervals of return time such that the speed of light multiplied by one time interval equates to a range extent at the target that is much smaller than the total extent or projection of the target along the sensor-to-target vector. Each such range or time interval equivalent is a range "bin."

**hypothesis** From Webster, a proposition or supposition tentatively accepted to provide a basis for further investigation (e.g. in our case, a choice of *a priori* likely parameter sets for testing against observed measurements from an unknown object or parameter set).

**hypothetical aspect angle sphere** A hypothetical sphere of unit radius, centered on the origin of the object body frame and fixed with respect to the body frame axes. Provides a physical representation for the concept of a two-dimensional *aspect angle path, region* or *space* (these terms defined above). See Fig. 1.2.

**Kalman filter** A class of mathematical estimators designed to make optimal estimates of quantities (states) for which time rate of change (propagation) can be described as linear equations driven by white Gaussian noise of known statistics and deterministic inputs, and for which measurements are available which can be described as a linear combination of some or all of the states and inputs, plus additive white Gaussian noise of known statistics. The term "optimal" is with respect to a number of possible criteria for judging estimator optimality: the estimate is the mean, mode, and median of the conditional density of the estimated variable, conditioned on available measurements; the estimate is the minimum mean-square error estimate; and so on (see [153:231-236]).

**kinematic** Having to do with the branch of mechanics (kinematics) that deals with aspects of motion (position, velocity, acceleration, and so on in translational *and* rotational degrees of freedom) *apart from considerations of mass or force* (see [144]). Generally

used in multisensor fusion to refer to measured quantities related directly to object translation, e.g., sensor-object relative position, or range and spatial angle quantities, and their derivatives.

**likelihood function, generalized likelihood function** The classical likelihood function  $p(\mathbf{z} | \omega_i)$  is the probability or probability density function, if the latter exists, of obtaining a particular measurement  $\mathbf{z}$ , given that we are observing an object of class (i.e., state and parameter set)  $\omega_i$  – with the *maximum likelihood* classifier assigning an unknown object to the class for which this likelihood function is maximized. A generalized likelihood function  $L$  is a function of  $\mathbf{z}$  and  $\omega_i$  defined for some identification problem, ideally such that the maximum likelihood function value for measurements from a true system with states and parameters  $\omega_i$  is obtained from the generalized likelihood function optimized for that set of states and parameters, so that a correct state and parameter set identification can be made. The likelihood function value need not provide a probability measure or analogous quantity (e.g., a Dempster-Shafer mass). See Sect. 2.6.

**morph** A short form of the expression *morphological transform* – generally, a change in the shape of a two- or three-dimensional physical object (see [22]).

**motion warping** As defined for this research, the process of using dynamic programming-based sequence comparison techniques to compare (1) a sequence of observed feature vectors from a object of unknown class with (2) object library models, using measurements of object kinematics and dynamic restrictions for each class known (approximately) *a priori*, to determine the most likely object class to have generated the observed sequence of feature vectors and kinematic measurements.

**multisensor fusion** From [218:1] (given there as a definition for *data* fusion, but equally applicable for *multisensor* fusion), “a multilevel, multifaceted process dealing with the detection, association, correlation, estimation, and combination of data and information from multiple sources to achieve refined state and identity estimation, and complete and timely assessments of situation and threat.” As employed in this research, the term *multisensor* may refer to different elements of information obtained

from the same piece of physical hardware: e.g., object position measurements *and* HRR signatures obtained from the same radar assembly.

**nonkinematic** Used in multisensor fusion to refer to measured object quantities not directly indicative of (and not having the dimensions of) translation or rotation states and higher derivatives thereof, e.g., feature observables, sensor signatures, etc.

**nonparametric** Decision theoretic pattern recognition or classification techniques which do not assign a *probability* measure of class membership to points based on their position in feature space, making instead a class membership decision for a given measurement based on the *distance* (in some defined metric) in feature space from the measurement to points corresponding to known classes. Also used to refer to techniques for estimating the probability density function of a process without *a priori* knowledge as to the form (Gaussian, Chi-square, etc.) of the density.

**object-to-sensor vector or unit vector** The vector (or corresponding unit vector) assumed to originate at the origin of the object body frame and terminate at the sensor aperture.

**occlusion, occluded** An occlusion is an obstruction in the line of sight between an object and a location which prevents all or part of the (*occluded*) object from being observed at the given location.

**off-nominal** A term used in this research to refer to errors between (1) the true object-sensor aspect angle at any point in time and (2) the corresponding object-sensor aspect angle as estimated from measured object kinematics and an assumption of object class, such that the error so described *does not* lie on the aspect angle path for both true and estimated aspect angle sequences.

**one-to-one** From [6:36], a term used to describe a function  $F$  for which each element in the range  $F(S)$  of the function corresponds to only one element in the domain  $S$  [6:36].

**on-nominal** A term used in this research to refer to errors between (1) the true object-sensor aspect angle at any point in time and (2) the corresponding object-sensor aspect angle as estimated from measured object kinematics and an assumption of

target class, such that the described error lies along the aspect angle path for both true and estimated aspect angle sequences.

**onto** From [6:35], a word describing the relationship between a function  $F$ , its domain  $S$ , and a set  $T$  containing the range  $F(S)$  of the function, in which  $F(S) = T$ .

**parameter** A quantity of interest to an estimator, fundamental to defining the behavior of a system over time as for "state" (defined below), but which, unlike states, is assumed not to change significantly over time periods of interest.

**parametric** A term used to describe decision theoretic classification or pattern recognition techniques which explicitly assess some measure of the probability of observed events (i.e., following the second meaning given below for statistical pattern recognition). Also used to refer to techniques for estimating the probability density function of a process, where the form (Gaussian, Chi-square, etc.) of the density is known *a priori*.

**pose estimate** An estimate of the aspect angle presented by some object, as viewed by a sensor.

**PSRI** An abbreviation for "Position-, Scale-, and (In-Plane) Rotation-Invariant." This phrase means that the value so termed, corresponding to an observation of some object by some sensor, will be the same, (1) independent of *position* changes normal to the sensor-object vector (so long as the object remains in the field of view), (2) independent of *scale* changes as from changes in sensor-object range or magnification, and (3) independent of rotations of the sensor relative to the object about the sensor-object vector only. This term does *not* imply invariance of the described value with respect to object *aspect angle* changes. Also, quantities that are theoretically PSRI may not be so in practice, due to factors like pixel orientation and size.

**radar cross section (RCS)** From [16:A-14]: a measure of the reflective strength of a radar target; usually represented in square meters (or decibel square meters – dBsm), and defined as  $4\pi$  times the ratio of (1) the power per unit solid angle scattered in a specific direction to (2) the power per unit area in a plane wave incident on the scatterer from a specified direction.

**range sweep** The return or output realized by reflecting one pulse from a high range resolution radar against a target, expressed as radar cross section in each of many "range bins" in some multiple-bin interval along the sensor-target vector (may also refer to the summed results from many such pulses).

**residual** the difference between measurements observed and predicted (generally, by an estimator) for some measurement event (see Sect. 2.3.1.1).

**roll angle** An Euler angle about the velocity frame  $\vec{x}_v$  axis, taken prior to sideslip angle and angle of attack, first in the sequence of Euler angles required to rotate the velocity frame into the body frame (see Fig. 5.24). This is not the same as *bank* angle (see above).

**segmentation** A general term for the process by which a sensor (usually an imaging sensor) system separates potential objects from probable background clutter, generally prior to processing the separated or *segmented* objects for recognition or discrimination, although segmentation may be performed in combination or iteratively with other processes.

**sideslip angle** The angle, generally and in this work denoted  $\beta$ , between an aircraft's wind vector (or velocity frame unit vector  $\vec{x}_v$ , for an atmosphere at rest with respect to the inertial frame as assumed in this work) and the (perpendicular) projection of that vector on the body  $\vec{x}_b$  -  $\vec{z}_b$  plane. See Fig. 5.23.

**slide distance** A distance defined by Algorithm Q (see above) which attempts to quantify the difference between two high resolution radar range sweeps, in accordance with a metric defined by General Dynamics (see Sect. 2.2.3).

**smoother** An estimator which makes estimates of some quantity (state, etc.) at some time, based or conditioned not only on measurements *prior* to that time, but also on measurements taken *after* that time (see Sect. 2.3.1.2).

**state** From [153:26]: the state of a system at any time  $t$  is a minimum set of values  $x_1(t), \dots, x_n(t)$  (an  $n$ -dimensional vector), which, along with the input to the system for all time  $\tau$ ,  $\tau \geq t$ , is sufficient to determine the behavior of the system for all  $\tau \geq t$ .

**statistical pattern recognition** A term used to refer either to decision theoretic (see above) pattern recognition concepts in general, or, (more precisely) to decision theoretic methods which define a probability or classical likelihood of class membership based on the location of a feature observable measurement in the feature space (the latter also called "parametric" classification methods).

**structural pattern recognition** A family of pattern recognition concepts which assign class membership based on the type, number, and, in some sense, relationships or structure between observed features for some object of unknown class, and features for known classes.

**sup** An abbreviation for supremum, or least upper bound of a set [6:9]. If the set has a maximum element, that element is also the supremum.

**syntactic pattern recognition** An alternative name for *structural pattern recognition* (see above), reflecting the language-based origin of many techniques in this area [90].

**target acquisition and tracking** The process of identifying a potential target using a sensor system and following the motion of that target over time.

**warping path** A particular sequence of associations between elements of two feature observable sequences. Due to continuity constraints, or rules for the associations, the allowable sequences of associations have the appearance of paths through the warping path region or "space" (see Fig. 2.8).

**warping path cost** The total cost or distance associated with a particular sequence of associations between elements of two feature observable vector sequences. The path having minimum warping path cost, subject to various rules for the associations, is taken to identify the expansions, compressions, insertions, and deletions which make the two sequences most similar to each other (see Eqn. (2.29)).

**warping path region or "space"** The finite set of all possible associations between elements of two finite sequences of feature observable vectors, from which sets of associations may be defined under applicable rules to define warping paths. These spaces are illustrated for (1) warping of two one-dimensional sequences in Fig. 2.8 (producing a "two-dimensional" warping path region or space) and for (2) warping

of a one-dimensional observed sequence against a two-dimensional region of possible sequences in Fig. 2.9 (producing a “three-dimensional” warping path region or space).



## Appendix B. Background Information on Pattern Recognition

This appendix is intended to provide a tutorial overview of pattern recognition for those without a background in the field. It should be read following the brief overview in Sect. 2.2.

### B.1 Taxonomy of Pattern Recognition Concepts for 3-D Objects.

The problem with taxonomies for pattern recognition is that every author in the field has his or her own. More specifically, pattern recognition is a *multidimensional* topic in the mathematical sense of the word, and each author makes distinctions along his or her preferred directions in the space of all possible pattern recognition concepts. The following “multidimensional” taxonomy is fundamentally based on the works of the late renowned K.S. Fu [90, 89, 88], Miclet [161] and Tou and Gonzalez [212], and draws from the discussion on concepts for “recognition, tracking, and pose estimation of arbitrarily shaped 3-d objects...” given by Gottschalk *et al.* [102], further supplemented by material from Duda and Hart [72], Fukunaga [91, 92], Pratt [178], Nasr [169:111–139] and others [55, 76, 176].

Fu [90] sets the fundamental distinction in pattern recognition as between “decision theoretic” and “syntactic” (or “structural”) methods. Decision theoretic classification methods include the classical nonparametric (feature space distance-based *discriminant* or *decision function*) and parametric (probabilistic, ideally Bayesian) classifiers. Miclet [161] similarly sets the fundamental distinction as one between “statistical” or “syntactic/structural” methods, including, as does Fukunaga [91, 92], both parametric and nonparametric classifiers under the heading of “statistical.” By contrast, Tou and Gonzalez [212] would agree generally with Fu’s taxonomy, but use the term “mathematical” rather than “decision theoretic,” and add a third category, “heuristic” (ad hoc procedures, often based on human experience, including artificial intelligence methods). In contrast to Miclet and Fukunaga, Tou and Gonzales would consider only parametric classifiers to be “statistical” in nature. Note that we specifically distinguish the task of *classification* from (probability) *density estimation* – density estimation also can be parametric or non-

parametric, respectively, depending on whether or not the form of the density (Gaussian, Poisson, etc.) is known *a priori*.

In general, it can be said that decision theoretic/mathematical, or (in the broad sense of Miclet and Fukunaga) statistical classification concepts are those which require measurement or calculation of some quantity or quantities (features) for all known object classes to establish a library or map in the *feature space* (equivalently, in the terminology established in Chapter I, the *feature observable space*). The recognition process then seeks to establish class membership for an object of unknown class, based on the nearness (in some sense) of the object's measured features to elements in the library or points on the map. Moreover, the parametric, or "true" statistical classifier, provides a relative measure or estimate of the *probability* of the object's membership in any given class.

In structural or syntactic pattern recognition, on the other hand, we seek to define a "grammar", whereby each object class corresponds to a particular order of "pattern primitives" – subpatterns which are readily recognized by the classifier. Pattern primitives in a syntactic classifier could be identical to the features used in some decision theoretic classifier – the key difference is the syntactic classifier's concern with *order of presentation*. A syntactic description of an object is generally some form of string or sequence, and recognition of an unknown object is reduced to comparing its observed pattern primitive sequence with sequences for known classes. Syntactic methods commonly make use of analysis techniques and terms applied to the study of human and animal languages. Significantly for our later purposes, Miclet [161] noted the association between syntactic pattern recognition and dynamic programming sequence comparison techniques for speech processing (Fu was undoubtedly aware of this relationship, but did not discuss it in detail in works known to this author, cited above). This association is further discussed in Sect. 2.4.5.

The approaches of Fu, Miclet, and Tou and Gonzalez can be compared with that discussed by Nasr [169:111], who saw from his multisensor fusion/object recognition perspective, a fundamental distinction between "statistical" and "model-based" approaches. There is an important dichotomy here in approach between Nasr and his community with respect to the former authors. Nasr is concerned with *information storage* techniques – his

pattern recognition processing techniques are all basically decision theoretic or heuristic (artificial intelligence) in nature. To Nasr, a "statistical" pattern recognition system is one which maintains a library *database*, or mapping, or set of decision surface parameters for feature information recorded *a priori* for objects, aspect angles, and conditions of interest. This is certainly the original and most common way to store data for decision theoretic classifiers, but the association between data storage method and classification method is not exclusive from either direction.

Similarly, the potential exists for confusion between the syntactic or structural approach of Fu *et al.* and Nasr's models in the sense that structural approaches often specify some "model" as the source of the distinctive sequence. But to Fu *et al.*, a model would most often be an abstraction – an automaton [88, 90, 212, 70] or one of its analogs (e.g., a hidden Markov model [176, 68]), generating a sequence of pattern primitives with a distinctive structure, while Nasr's model is a software simulation of a real three-dimensional object like a tank or an aircraft, and the physics that produce observables of interest. Moreover, the output of Nasr's model would not generally be a characteristic sequence, and might well be a predicted measurement vector or function no different in form from those maintained in the database of his "statistical" classifier.

Gottschalk [102] on the other hand, working from the classical image recognition perspective, made the distinction between pattern recognition *techniques* using (1) global descriptors or (2) point correspondences as we will discuss below. Briefly, global descriptors are mathematical quantities that are defined in general by the entirety of an observed object at any particular aspect angle. If any element of the object (as observed or measured by the sensor) changes, then the global descriptor value may change. Alternatively, point (or locus of points) correspondence methods require the classifier to identify the presence or absence of some particular point, line segment, entity or combination thereof on the object, such as one would perceive when viewing the object, in comparison with *a priori* data on likely object classes. Gottschalk's view of data storage by library database or model representation is similar to Nasr's.

Relating Gottschalk's work to Fu *et al.* and Nasr, we note that global descriptors are commonly used as features in decision theoretic/mathematical classifiers, while point

correspondence systems could be classed as either heuristic, syntactic, or decision theoretic, depending on the operations performed using the point-based information. To Gottschalk's two distinctive techniques for image recognition we will add and distinguish those that use correlations (which could be considered a kind of pointwise correspondence). Correlations should probably be considered as a very special form of decision theoretic classifier, since they can rather directly provide a numerical "distance" value indicating similarity or, perhaps with proper design and training, probability of class membership. Inherently, however, the correlation process considers structural information.

A particular caution is in order at this point regarding the use of the term "feature." In discussions on global descriptors, the term "feature" is generally synonymous with the descriptor itself – a numerical quantity derived from a sensor, a particular value of which defines a particular location in some "feature space." In discussions on correspondence methods, however, a feature is that particular point, line segment, entity, or combination thereof on the object – sometimes called a "*landmark*" [102] – which we desire to place in correspondence with the like entity on the correct object library representation.

## *B.2 Decision Theoretic Object Recognition.*

Combining the decision theoretic concepts of Fu *et al.* [90] with the terminology for data storage techniques used by Nasr [169], we now contrast the classical nonparametric and parametric (Bayesian and maximum likelihood) approaches to object recognition as widely used in the multisensor fusion community.

*B.2.1 Decision Theoretic Methods – Survey.* Recall that parametric or probabilistic decision theoretic classifiers and the clustering of features in feature spaces for *a priori* known objects and aspect angles were discussed in Sect. 2.2.1. By comparison, in a typical *nonparametric* classifier, we use the same *a priori* information by measuring the features of an unknown object, and then find the cluster closest *in some distance sense* to the unknown object's feature set or vector. The factors which defined this cluster are taken then to indicate the unknown object's class and orientation (often a discretized value reflecting an orientation within some partitioned aspect angle interval or "bin"). Clearly,

the choice of metric and definition of "close" in our multidimensional feature space are critical, as are the centroid-to-centroid separation and dispersion of the clusters.

One definition of distance or closeness calls for each object class/aspect combination to be represented by a cluster centroid (mean) or single prototype point, so that the shortest measurement-to-prototype distance "wins" (i.e., the unknown object is taken to belong to the class of the nearest prototype) [212:77]. Another definition of closeness is established by defining *decision surfaces* or *functionals* in the feature space to separate the clusters – these *functionals* operate on observed feature values (vectors) to produce scalars – measures of distance between the observed values and the surfaces. Precise locations for decision surfaces may be defined automatically through a "supervised training" process – providing the classifier with a number of observations labelled as to which object class produced each, and allowing the decision surface parameters to converge to appropriate values.

If the clusters are of different dispersion or "diameter," but completely separated (in the sense that their convex hulls do not intersect), we will probably prefer to define a system of separating hyperplanes (linear functionals [170]) or other decision surfaces such that the position of a measured feature vector relative to the surfaces will associate it with a particular cluster [212:40–48]. If the clusters overlap but their component points are equally likely to occur overall (or if the relative number of points in the clusters reflect the *a priori* likelihood of the conditions associated with each cluster), we may prefer to use a "K-nearest neighbor" technique – assigning the unknown object to the pattern class which has the largest number of *a priori*-defined data elements in the total set of "K" data elements that are closest to the unknown object feature vector [212:81–83]. As an example of how these various approaches relate, observe that if each cluster has only one point, then the (single) nearest neighbor technique is equivalent to the "nearest prototype" method.

Relating parametric to nonparametric approaches, we may observe that for two clusters with Gaussian distributions, the decision surface, or locus of points in feature space that defines the boundary between points with higher probability of belonging to one cluster versus belonging to the other, is described by a quadratic (more properly, for higher dimensions, hyper-quadratic) equation – defining the *quadratic classifier*. If the distributions (covariances) of the two clusters are identical, the surface reduces to a hyperplane

normal to and bisecting the line segment joining the two cluster centers, and the decision logic is then identical to that of a linear classifier (i.e., a nonparametric decision made with a separating hyperplane).

All nonparametric classifier techniques result generally in some object-to-class assignment, but they do not provide all of the information we would like to have – in particular, we need a metric of class membership likelihood that can be combined with results from other classifiers, with each result weighted optimally according to our confidence in the individual classifier. If we have several separate object recognition classifiers providing answers which we must fuse to derive a final answer, this kind of relative information is critical – otherwise, we may be reduced to a “voting” system, in which each classifier has only one “vote” (although votes may be weighted heuristically based on our confidence in the answer given by a particular *classifier*).

The probability measures provided by parametric classifiers, however, do provide a metric suitable for comparison with other classifiers. The improved utility of information and decision optimality provided by a (true) statistical or parametric classifier over a nonparametric classifier in multisensor fusion and object recognition applications has brought about a general preference for parametric classifiers over nonparametric classifiers for associating measurements with *a priori* object feature information, commonly stored in a multidimensional feature space representation or library database. This habitual association, coupled with the terminology differences between authors noted above, commonly leads to the somewhat ambiguous application of the term “statistical” to the classification technique, the data storage technique, or the combination of the two.

**B.2.2 Statistical Library Approaches.** The “statistical library” or database/feature space mapping is the original and most common method for storing *a priori* pattern class information and implementing decision theoretic classification concepts. Considering this database as a mapping of recorded feature values from known classes into our feature space, decision theoretic concepts like separating hyperplanes are readily envisioned. All such libraries are built by measuring or calculating and recording values of our chosen features for selected values of object class, orientation, and other conditions (operating

state, background, etc.). In recent object recognition research reviewed by the author, these features generally have been global; e.g., see [42, 43, 44, 94, 101, 169, 8]. Measurements may be extracted from actual objects or models, and preserved as such or reduced to decision surface functions, but the object or model form is not preserved as such in the database.

An increasingly popular alternative to classical decision theoretic techniques with "statistical" library databases is provided by artificial neural net (ANN) methods [185, 219, 174] and other "trainable" classifiers. It has been recognized for some time that "multi-layer perceptron, feedforward" neural nets and a variety of related classifier concepts can be "trained" in a "supervised" fashion, so that a trained net provided with an unlabelled observation will output or indicate the appropriate object class. The neural net is both a data storage technique and a classifier.

The significant advantage of neural nets and similar trainable classifiers over classical clustering and statistical classifiers is simply that the designer expends no effort in defining clusters, decision surfaces, or object class probability densities in his or her chosen feature space: given an adequate classifier structure to begin with, the classifier's parameters (generally) converge automatically to or "learn" appropriate values during the training process. In particular, unlike classical nonparametric training approaches, neural nets are generally not restricted to a particular functional form (linear, quadratic, etc.) of decision surface – the net constructs arbitrary decision surfaces as required from elemental functions in the net structure.

Originally, the output from the typical multilayer perceptron feedforward net was taken only as indicating the "most likely" or "nearest neighbor" object class (source) for the observed data (a nonparametric classification). Recently, however, Ruck [190] has shown that this net approximates a Bayesian statistical classifier under certain conditions, so that the relative magnitude of output values from the nodes representing the various object classes may be taken as proportional in some sense to values of  $p(\omega_i | \mathbf{z})$  in Eqn. (2.1).

The problem, however, with all statistical library approaches is their inflexibility to changing environmental, background, operational, and other conditions that affect object

signatures. By including all possible such conditions in defining our databases of feature values for each object/aspect, or in estimating the classical likelihood  $p(\mathbf{z} \mid \omega_i)$  for each  $\omega_i$ , or in training a neural net, we may increase ambiguity, and may decrease the probability of making a correct decision in any one particular set of circumstances. Building separate databases or neural nets for each likely set of circumstances has its own pitfalls, principally data storage problems and the unavoidable fact that, when approximating a multidimensional continuous function (i.e., the set of all possible object scenarios) with values at a finite number of discrete points, we will effectively never have the correct combination required for a given implementation.

*B.2.3 Model-Based Approaches.* Recognition of (1) the problems with statistical library-based approaches and (2) the greater capability afforded by modern data processors has led the object recognition community to a general agreement that model-based systems offer a more promising approach. "Model-based," in the sense implied by Nasr [169] and others [218] implies that the recognizer carries some form of 3-D representation of each potential object and, accounting for object aspect, operation, environment, background, and other conditions, calculates on-line in near-real time what the sensor should see, providing a basis for comparison with actual measurements.

The level of model fidelity is completely open to the designer. Systems discussed in current literature vary from use of "wire frame" or stick models to exceptionally complicated models that consider, for example, heat flow between vehicle components (e.g., see [169]), or multiple reflection of radar waves in engine cavities [21]. Certainly, suitability of very highly complicated models for on-line implementation remains an open issue.

Fundamentally, model-based approaches seek to find an object model and (generally, in a 3-D model) orientation that define the "nearest neighbor" to the given observation. Thus they are most generally (but not exclusively) suitable to nonparametric classification.



### *B.3 Pattern Recognition Using Global Descriptors.*

As noted above, global descriptors are numerical quantities that are defined by the entirety of an object at any particular aspect angle. If any element of the object as measured by the sensor changes, then the global descriptor value may change.

Generally, the term "global descriptor" refers to some value extracted from a 2-D image. Resolution and identification of specific local entities on the object are not important in most global descriptor implementations. The fundamental attraction and yet weakness of many classical image-extracted global descriptors is that they are largely or completely defined by the object silhouette. However, dependence on the "whole image" makes these global descriptors sensitive to errors by occlusion and segmentation (see Appendix A for definitions). Another attraction of global descriptors is that many are position-, scale-, and (in-plane) rotation-invariant (PSRI). In terms of pattern recognition for 3-D objects from arbitrary aspect angles, this means that the global descriptors are nominally (excepting noise, environment, and background effects) a function only of object type and two object body-to-sensor angles, say azimuth and elevation. Target scale (range) and rotation angle ("roll") about the sensor-object vector are irrelevant in theory (practical factors, like pixel size and orientation, may cause problems in practice). Three particularly popular forms of global descriptors are discussed in the following subsections.

*B.3.1 Moments.* Analogously with the concept of two-dimensional mass moments for a flat physical object, we can define image moments for a segmented object. The moment calculation may treat each pixel within a given segmented contour as having equal "mass," or it may weight each pixel according to some additional information. For example, pixels in IR imagery may be weighted according to their image intensity – thus a "hot" pixel has more "mass" than a "cold" one.

Starting with these fundamental "physical" moments, Hu [111] derived a set of PSRI features, commonly called "Hu moments" that have seen extremely wide application [225]. See either of [111, 178] for a complete listing.

*B.3.2 Fourier Descriptors.* Fourier descriptors [98, 178, 217, 101] are a class of frequency domain descriptions for the silhouette of a segmented object. For example, we may take the Fourier series expansion of the curvature of the silhouette, curvature being defined as the derivative of the tangent angle with respect to length along the silhouette curve. Clearly, the curvature is periodic and real as we make multiple traverses of the silhouette – thus we can find the Fourier series expansion for the curvature, and extract a finite number of Fourier coefficients, which are the descriptors. Fourier descriptors are subject to “noise” from poor segmentation and occlusion, since these error sources affect silhouette most immediately. Note that Fourier descriptors are not to be confused with low-frequency Fourier spatial frequency components extracted by digital signal processing or optical correlations, since spatial frequency involves image factors other than the segmented silhouette (see Sect. B.4).

*B.3.3 Miscellaneous Global Descriptors.* A variety of other features have been defined and used for object recognition with imaging sensors. These include “complexity” (ratio of number of edge pixels to number of internal pixels for some segmented region), height-to-width ratio, brightness (intensity), “texture”, and so on (see [218:99], for example). For the most part, these are PSRI (position, scale, and rotation-invariant) quantities. Their utility and significance in segmentation and recognition have been investigated in several studies, e.g. [186, 190, 226, 225].

#### *B.4 Pattern Recognition Using Correspondences.*

Correspondence methods require the classifier to identify the presence or absence of some particular point, line segment, object or combination thereof (“features” or “landmarks” for this type of classifier) on the object, in comparison with *a priori* data on likely object classes [102, 110]. Correspondence classifiers are increasingly model-based, but classification techniques may be heuristic, syntactic, or decision theoretic. The key to success in a correspondence classifier is making the object-model-feature to observed-object-feature association – in general this calls for higher resolution on the object surface than required

by global descriptor-based systems. Also, correspondence-based classifiers are generally not position-, scale-, and (in-plane) rotation-invariant (PSRI).

#### *B.5 Pattern Recognition Using Correlations.*

Target-image to library-image (template) correlations can be performed using 2-D digital image processing or by (much faster) optical processing using the Fourier transform properties of optics [47, 58, 93, 98, 99, 103, 158, 178, 188, 232]. In general, correlation systems are not position, scale, and (in-plane) rotation-invariant, but at least one technique achieves these attributes, by transforming the object and test images as shown in Kobel and Martin [127], using the method of Horev [109]. Range "maps" obtained by laser radar measurement of unclassified targets could be matched to known targets for varied aspect angles using forms of 3-D correlation.

Since correlations are extremely sensitive to sensor-object aspect angle [103] and object operating state, they are perhaps best accomplished using object models which can be oriented and adjusted for operating conditions, etc., to produce the "best" correlation with an input object. The model producing the highest overall correlation value is taken to indicate the object class. Correlations can be performed using a large library of object representations over various aspect angles, but this approach is increasingly less favored due to data storage requirements.

#### *B.6 Summary*

In combination with the ideas presented in Sect. 2.2, this appendix has presented a brief overview of key concepts in pattern recognition, particularly as applied to military targets. For further information in this area, the author recommends that the reader scan the bibliography for titles of interest. General references [72, 212, 92, 90, 169, 178, 70, 174] are appropriately titled. Other sources relating to particular sensors and their phenomenology may also be of interest, and can be identified by their titles.

## Appendix C. Detailed Equations for Estimation and Tracking Algorithms

### C.1 Introduction

This appendix primarily presents the dynamics (state) and measurement equations for the Kendrick/Maybeck/Reid [120, 121], Andrisani/Kuhl/Gleason [5], and Swarder/Hutchins [208, 209] estimators. The equations are presented here using their original variables, for which definitions are provided.

A less detailed description of the Bishop estimator is also provided. This tracking algorithm was briefly mentioned in Sect. 2.3.1.4.

Finally, the equations for fixed point and fixed lag smoothing are given as in [154:16-17], with minor comments regarding their implementation for this research. Additional results for scenarios like those in Chapter V are also presented.

### C.2 The Kendrick / Maybeck / Reid Estimator

#### C.2.1 Kinematic Filter.

**C.2.1.1 Kinematic Filter State Equations.** Using the form of Eqn. (2.14), the state equations for the kinematic filter in the Kendrick *et al.* estimator are:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p}_{t/a_N} \\ \dot{p}_{t/a_E} \\ \dot{p}_{t/a_D} \\ \dot{V}_{t/a_N} \\ \dot{V}_{t/a_E} \\ \dot{V}_{t/a_D} \\ \dot{\delta a_N} \\ \dot{\delta a_E} \\ \dot{\delta a_D} \\ \dot{\epsilon} \end{bmatrix} = \begin{bmatrix} V_{t/a_N} \\ V_{t/a_E} \\ V_{t/a_D} \\ g(\alpha + \beta e^{\gamma \epsilon})(l_N)_N + \delta a_N - \dot{V}_{a/I_N} \\ g(\alpha + \beta e^{\gamma \epsilon})(l_N)_E + \delta a_E - \dot{V}_{a/I_E} \\ g(\alpha + \beta e^{\gamma \epsilon})(l_N)_D + \delta a_D + g - \dot{V}_{a/I_D} \\ -\frac{1}{\tau_N} \delta a_N \\ -\frac{1}{\tau_E} \delta a_E \\ -\frac{1}{\tau_D} \delta a_D \\ -\frac{1}{\tau_\epsilon} \epsilon \end{bmatrix} + \mathbf{G} \begin{bmatrix} W_{V_N} \\ W_{V_E} \\ W_{V_D} \\ W_{\delta a_N} \\ W_{\delta a_E} \\ W_{\delta a_D} \\ W_\epsilon \end{bmatrix} \quad (\text{C.1})$$

where:

$p_{t/a_{N,E,D}}$  = position of the target relative to the attacker in inertial frame coordinates, i.e., with components taken along the North, East, or Down axes of an earth-surface inertial or navigation frame. Dot notation denotes time rate of change of the indicated variable, as seen in the frame used for coordinatization.

$V_{t/a_{N,E,D}}$  = velocity of the target relative to the attacker, as observed from and coordinatized in the navigation (inertial) frame

$\delta a_{N,E,D}$  = acceleration of the target relative to the attacker, in addition to acceleration from normal load (lift), as observed from and coordinatized in the navigation (inertial) frame

$g$  = acceleration due to gravity

$A_N = g(\alpha + \beta e^{\gamma \epsilon})$  = the magnitude of normal load acceleration, modeled as a positive (non-zero) mean random process driven by the random variable  $\epsilon$

$\epsilon(t)$  = a random variable which drives  $A_N$  (strictly,  $\epsilon$  is a stochastic process)

$\alpha, \beta, \gamma$  = parameters peculiar to particular aircraft types and operating conditions

$e$  = the base of natural logarithms

$(l_N)_{N,E,D}$  = the north, east, and down components of the unit vector ( $l_N$ ) in the direction of normal load acceleration

$\tau_{N,E,D}$  = correlation times for first-order Gauss Markov models used to model accelerations other than that due to (nominal) normal lift

$\tau_\epsilon$  = correlation time for the first order Gauss Markov process modelling the behavior of  $\epsilon$

$\dot{V}_{a/I_{N,E,D}}$  = acceleration of the attacker (sensor) relative to the navigation (inertial) frame, coordinatized in the navigation frame (assumed available from the attacker's inertial navigation system – errors implicitly considered in noise parameters listed above)

$W_{V_N, V_E, V_D, \delta a_N, \delta a_E, \delta a_D, \epsilon}$  = appropriate continuous time (heuristically) zero-mean white Gaussian process driving noises, with some strength  $Q(t)$  as defined in Sect. 2.3.1

and :

$$\mathbf{G} = \begin{bmatrix} \mathbf{0}_{3 \times 7} \\ \mathbf{I}_{7 \times 7} \end{bmatrix} \quad (\text{C.2})$$

*C.2.1.2 Kinematic Filter Measurement Equations.* Using the form of Eqn. (2.15), the measurement equations for the kinematic filter in the Kendrick *et al.* estimator are:

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} r \\ \eta \\ \sigma \\ \dot{r} \\ \dot{\eta} \\ \dot{\sigma} \end{bmatrix} \quad (\text{C.3})$$

where:

$r$  = attacker-to-target range:

$$r = [(p_{t/a_N})^2 + (p_{t/a_E})^2 + (p_{t/a_D})^2]^{\frac{1}{2}} \quad (\text{C.4})$$

with corresponding rate of change:

$$\dot{r} = \frac{(p_{t/a_N} V_{t/a_N}) + (p_{t/a_E} V_{t/a_E}) + (p_{t/a_D} V_{t/a_D})}{[(p_{t/a_N})^2 + (p_{t/a_E})^2 + (p_{t/a_D})^2]^{\frac{1}{2}}} \quad (\text{C.5})$$

$\eta$  = (azimuth) angle between the projection noted above and local north at the attacker's position:

$$\eta = \arctan \left( \frac{p_{t/a_E}}{p_{t/a_N}} \right) \quad (\text{C.6})$$

with corresponding rate of change:

$$\dot{\eta} = \frac{(p_{t/a_N} V_{t/a_E}) - (p_{t/a_E} V_{t/a_N})}{(p_{t/a_N})^2 + (p_{t/a_E})^2} \quad (C.7)$$

$\sigma$  = (elevation) angle between the target/attacker vector and the vertical projection of this vector onto the local horizontal plane:

$$\sigma = \arctan \left[ \frac{-p_{t/a_D}}{[(p_{t/a_N})^2 + (p_{t/a_E})^2]^{\frac{1}{2}}} \right] \quad (C.8)$$

with corresponding rate of change:

$$\dot{\sigma} = \frac{p_{t/a_D} [(p_{t/a_N} V_{t/a_N}) + (p_{t/a_E} V_{t/a_E})] - V_{t/a_D} [(p_{t/a_N})^2 + (p_{t/a_E})^2]}{[(p_{t/a_N})^2 + (p_{t/a_E})^2 + (p_{t/a_D})^2][(p_{t/a_N})^2 + (p_{t/a_E})^2]^{\frac{1}{2}}} \quad (C.9)$$

$\mathbf{z}(t_i)$ , the measurement at time  $t_i$ , is modelled as the sum of  $\mathbf{h}[\mathbf{x}(t_i)]$  from above (found using truth state values) and a vector of discrete time zero mean white Gaussian noise  $\mathbf{v}(t_i)$ , with an appropriate covariance  $\mathbf{R}(t_i)$

and all other variables are defined above.

### C.2.2 Aspect Angle Filter.

**C.2.2.1 Aspect Angle Filter State Equations.** Using the form of Eqn. (2.5), the state equations for the aspect angle filter in the Kendrick *et al.* estimator are:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \\ \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \theta \\ \phi \\ \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_\psi \\ w_\theta \\ w_\phi \end{bmatrix} \quad (C.10)$$

where:

$\psi$  = is azimuth angle, first of three Euler angle rotations taken to carry the navigation frame into the body frame (see [79:112–113])

$\theta$  = is elevation angle, second of three Euler angle rotations taken to carry the navigation frame into the body frame

$\phi$  = is bank angle, last of three Euler angle rotations taken to carry the navigation frame into the body frame

$w_{\psi,\theta,\phi}$  = appropriate continuous time (heuristically) zero-mean white Gaussian process driving noises, with some strength  $\mathbf{Q}(t)$  as defined in Sect. 2.3.1

**C.2.2.2 Aspect Angle Filter Measurement Equations.** Using the form of Eqn. (2.6), the measurement equations for the kinematic filter in the Kendrick *et al.* estimator are:

$$\mathbf{z} = \begin{bmatrix} \tilde{\psi}_s \\ \tilde{\theta}_s \\ \tilde{\phi}_s \\ \tilde{\psi}_k \\ \tilde{\theta}_k \\ \tilde{\phi}_k \end{bmatrix} \quad (\text{C.11})$$

where:

$\tilde{\psi}_s$  = imaging sensor-derived azimuth angle

$\tilde{\theta}_s$  = imaging sensor-derived elevation angle

$\tilde{\phi}_s$  = imaging sensor-derived bank angle

$\tilde{\psi}_k$  = kinematically-derived yaw angle (an estimate, or pseudo-measurement):

$$\tilde{\psi}_k = \arctan \left[ \left( \frac{V_y \cos(\alpha_t)}{|V|} + \frac{A_{N_y} \sin(\alpha_t)}{|A_N|} \right) \left( \frac{V_x \cos(\alpha_t)}{|V|} + \frac{A_{N_x} \sin(\alpha_t)}{|A_N|} \right)^{-1} \right] \quad (\text{C.12})$$



$\tilde{\theta}_k$  = kinematically-derived pitch angle (an estimate, or pseudo-measurement):

$$\tilde{\theta}_k = \arcsin \left( \frac{V_z \cos(\alpha_t)}{|V|} + \frac{A_{N_s} \sin(\alpha_t)}{|A_N|} \right) \quad (\text{C.13})$$

$\tilde{\phi}_k$  = kinematically-derived bank angle (an estimate, or pseudo-measurement):

$$\tilde{\phi}_k = \arctan \left[ \left( \frac{V_x A_{N_y} - V_y A_{N_x}}{|V| |A_N|} \right) \left( \frac{V_z \sin(\alpha_t)}{|V|} - \frac{A_{N_s} \cos(\alpha_t)}{|A_N|} \right)^{-1} \right] \quad (\text{C.14})$$

and  $\mathbf{H}$  is therefore given by:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (\text{C.15})$$

and all other quantities are defined above (note that  $|\dots|$  denotes magnitude of the quantity enclosed).

### C.3 The Andrisani / Kuhl / Gleason Estimator

The following equations are presented essentially as they appear in [5], with the exception that some changes were made to correct apparent errors. First, in the original source, the negative signs required for the state equation of the first order Gauss Markov process noises  $b_{x,y,z}$  were not present (it is possible, but not stated or conventional, that a negative correlation time was intended). Second, it is clear that the denominator in the arctangent expression in the sixth element of the measurement equation requires an exponent of  $\frac{1}{2}$ , which is not present in the original source. Third, review of [5] and [187] makes it clear that the term  $k_6 \nu_p$  is required in Eqn. (C.16), although it is not found as such in this equation as it appears in [5].

**C.3.1 State Equations.** Using the form of Eqn. (2.14), the state equations for the Andrisani *et al.* estimator are:

$$\begin{aligned}
\dot{\mathbf{x}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{b}_x \\ \dot{b}_y \\ \dot{b}_z \end{bmatrix} &= \begin{bmatrix} [-(I_{xx} - I_{yy})qr + k_1\beta + k_2p + k_3r] I_{xx}^{-1} \\ [-(I_{xx} - I_{zz})pr + k_6\nu_p + k_7\alpha + k_8q] I_{yy}^{-1} \\ [-(I_{yy} - I_{xx})pq + k_{11}\beta + k_{12}p + k_{13}r] I_{zz}^{-1} \\ p + q \sin(\phi) \tan(\theta) + r \cos(\phi) \tan(\theta) \\ q \cos(\phi) - r \sin(\phi) \\ [q \sin(\phi) + r \cos(\phi)] \cos^{-1}(\theta) \\ \left\{ \begin{aligned} &[\cos(\psi) \sin(\alpha) \cos(\theta) - \cos(\psi) \cos(\alpha) \cos(\phi) \sin(\theta) - \\ &\sin(\psi) \cos(\alpha) \sin(\phi)] \frac{L}{M} + k_{16}b_x \end{aligned} \right\} \\ \left\{ \begin{aligned} &[\sin(\psi) \sin(\alpha) \cos(\theta) - \sin(\psi) \cos(\alpha) \cos(\phi) \sin(\theta) + \\ &\cos(\psi) \cos(\alpha) \sin(\phi)] \frac{L}{M} + k_{16}b_y \end{aligned} \right\} \\ [-\sin(\alpha) \sin(\theta) - \cos(\alpha) \cos(\phi) \cos(\theta)] \frac{L}{M} + g + k_{16}b_z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ -\frac{b_x}{\tau} \\ -\frac{b_y}{\tau} \\ -\frac{b_z}{\tau} \end{bmatrix} +
\end{aligned}$$

$$\begin{bmatrix}
k_4/I_{xx} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & k_9/I_{yy} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & k_{15}/I_{zz} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & k_{16} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & k_{16} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & k_{16} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
w_3 \\
w_4 \\
w_5 \\
w_6 \\
w_7 \\
w_8 \\
w_9
\end{bmatrix}
\quad (C.16)$$

where:

$p, q, r$  = angular rates of the body frame relative to the inertial frame, along the body frame  $\vec{x}_b, \vec{y}_b$ , and  $\vec{z}_b$  axes as defined in Fig. 5.23.

$x, y, z$  = target-to-sensor distances along the respective inertial frame axes (first and second derivatives with respect to time as observed from the inertial frame denoted by one or two dots, respectively)

$\psi, \theta, \phi$  = inertial frame to body frame Euler angles as defined for the Kendrick *et al.* estimator.

$I_{xx, yy, zz}$  = moments of inertia about the body frame  $\vec{x}_b, \vec{y}_b$ , and  $\vec{z}_b$  axes, respectively.

$L$  = lift force magnitude,  $= \frac{1}{2}\rho(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)SC_{L\alpha}$ , where:

$S$  = equivalent lifting surface area, and

$C_{L\alpha}$  = the coefficient of lift

$\alpha$  = angle of attack as defined in Fig. 5.23, such that:

$\sin(\alpha) = C/\nu_p$ , where:

$\nu_p = (A^2 + C^2)^{\frac{1}{2}}$ , with:

$$A = \cos(\theta) \cos(\psi) \dot{x} + \cos(\theta) \sin(\psi) \dot{y} - \sin(\theta) \dot{z}$$

$$C = [\cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi)] \dot{x} + [\cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi)] \dot{y} + \cos(\phi) \cos(\theta) \dot{z}$$

$M = \text{mass}$

$g = \text{acceleration due to gravity}$

$b_{x,y,z} = \text{a time-correlated acceleration process noise, modeled as the output of a first-order Gauss Markov model}$

$\tau = \text{correlation time constant for the previously noted process noise}$

$k_{1,2,3,\dots,16} = \text{aircraft-class dependent constants, corresponding to quantities given in [187]}$

$w_{1,2,3,4,5,6,7,8,9} = \text{appropriate continuous time (heuristically) zero-mean white Gaussian process driving noises, with some strength } \mathbf{Q}(t) \text{ as defined in Sect. 2.3.1}$

**C.3.2 Measurement Equations.** Using the form of Eqn. (2.15), the measurement equations for the Andrisani *et al.* estimator are:

$$\begin{bmatrix} \phi_m \\ \theta_m \\ \psi_m \\ R \\ \eta \\ \epsilon \\ \dot{R} \\ \dot{\eta} \\ \dot{\epsilon} \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \\ (x^2 + y^2 + z^2)^{\frac{1}{2}} \\ \arctan\left(\frac{y}{x}\right) \\ \arctan\left[\frac{-z}{(x^2 + y^2)^{\frac{1}{2}}}\right] \\ (x\dot{x} + y\dot{y} + z\dot{z})(x^2 + y^2 + z^2)^{-\frac{1}{2}} \\ (x\dot{y} - y\dot{x})(x^2 + y^2)^{-1} \\ [z(x\dot{x} + y\dot{y}) - \dot{z}(x^2 + y^2)] [(x^2 + y^2 + z^2)(x^2 + y^2)^{\frac{1}{2}}]^{-1} \end{bmatrix} + \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \\ \nu_5 \\ \nu_6 \\ \nu_7 \\ \nu_8 \\ \nu_9 \end{bmatrix} \quad (\text{C.17})$$

where subscript m denotes measurement by a pose estimator, and

$\nu_{1,2,3,4,5,6,7,8,9}$  = discrete time zero mean white Gaussian noises with appropriate covariance  $\mathbf{R}(t)$  as defined in Sect. 2.3.1; evidently, Andrisani *et al.* chose to use " $\nu$ " rather than " $v$ " notationally as we have done for this form of variable.

$R$  (range),  $\eta$  (azimuth angle), and  $\epsilon$  (elevation angle) measurements and corresponding rate measurements are defined exactly as for the Kendrick *et al.* estimator.

and all other quantities are defined above.

#### C.4 The Swarder / Hutchins Estimator

Recognizing that most "pose" estimators assign the target to one of a finite number of discrete orientations (as discussed earlier in Sect. 2.2), say  $l$  in number, and that lateral acceleration can likewise be discretized, to say  $k$  in number, Swarder defines a Kronecker product space [38]  $\alpha \otimes \rho$  consisting of all possible orientation ( $\alpha$ ) and acceleration ( $\rho$ ) combinations. He then notes that transitions in this space occur according to the theory of *marked point processes*, as defined by Snyder [205]. Using this theory, Swarder *et al.* are able to define a differential equation which governs the estimate of an indicator for the presence of the system in any of the  $k \times l$  states:

$$d\hat{\phi}_t = \mathbf{Q}^T \hat{\phi}_t dt + (\text{diag}(\hat{\phi}_t) - \hat{\phi}_t \hat{\phi}_t^T)(\Lambda(1_k^T \otimes \mathbf{I}_l))^T \times \text{diag}(\hat{\lambda}_t^{-1}) d\sigma_t \quad (\text{C.18})$$

where (note that " $\times$ " denotes multiplication in the Swarder development, not vector cross product):

$\hat{\phi}_t$  = a vector of dimension  $k \times l$ , each element of which is a value from zero to one representing the probability that the true target is in that particular acceleration/orientation state

$\mathbf{Q}$  = a  $k \times l$  by  $k \times l$  transition rate matrix giving the probability rate of transitions between states - e.g., the probability that the target is in state 3 at some time  $t + dt$ , given that the target is in state 2 at time  $t$  is equal to  $q_{23} \times dt$ , where  $q_{23}$  is the element of the

matrix  $\mathcal{Q}$  in the second row and third column. Diagonal elements are handled differently – the probability that the target is in any state  $i$  at some time  $t + dt$ , given that the target is in state  $i$  at time  $t$ , is equal to  $(1 - q_{ii}) \times dt$

$diag(\hat{\phi}_t)$  = a  $k \times l$  by  $k \times l$  diagonal matrix, elements of which are the elements of the vector  $\hat{\phi}_t$

$\Lambda$  = is an  $l \times l$  matrix, the  $i$ - $j$ th (row-column) element of which is the expected rate of measurements showing target orientation in the  $i$ th aspect bin, if the target is in fact in the  $j$ th aspect bin

$(\mathbf{1}_k^T \otimes \mathbf{I}_l)^T$  = the transpose of the Kronecker product of the transpose of a  $k$ -dimensional vector of ones and an  $l \times l$  identity matrix, defining a  $k \times l$  by  $l$ -dimensional matrix

$diag(\hat{\lambda}_t^{-1})$  = a diagonal matrix of dimension  $l \times l$ , the elements of which are the inverse of the elements of the  $l$  dimensional vector defined by  $\Lambda \times (\mathbf{1}_k^T \otimes \mathbf{I}_l) \times \hat{\phi}_t$

$\sigma_t$  = a vector of length  $l$ , each element of which represents the number of observations with that particular aspect bin reading over the entire measurement history

$t$  = time

Like the Kalman filter state estimator, this estimator operates using a set of propagations and updates [208]. Propagations are given by:

$$\dot{\hat{\phi}}_t = \mathcal{Q}^T \hat{\phi}_t \quad (\text{C.19})$$

and updates by:

$$\delta \hat{\phi}_t = (diag(\hat{\phi}_t) - \hat{\phi}_t \hat{\phi}_t^T)(\Lambda(\mathbf{1}_k^T \otimes \mathbf{I}_l))^T \times diag(\hat{\lambda}_t^{-1}) \delta \sigma_t \quad (\text{C.20})$$

(where “ $\delta$ ” denotes an impulsive change in the indicated variables at measurement time, and all other variables are defined above). The measurement at any given time is the apparent discrete orientation value from the pose estimator. Note, as does Swarder, that although Eqn. (C.18) is presented in continuous-time form, for point process measurements,

where  $d\sigma_t$  is zero except at measurement updates, Eqns. (C.19) and (C.20) are a correct implementation of Eqn. (C.18). Finally, the orientation and acceleration values define an inertial acceleration which plays the part of a reasonably well estimated control input ( $\mathbf{u}(t)$ , in Eqn. (2.14)) for the state equation of an extended Kalman filter which uses conventional range and angle-derived kinematic measurements.

#### C.5 Nonlinear Kinematic Tracker Effort by Bishop.

Bishop's effort, referred to in Sect. 2.3.1.4, is based on work by Krener *et al.* [26, 87, 131, 132, 130, 45, 215], and shows that, for a particular target maneuver model, if a (one-to-one, onto,  $C^\infty$ , generally nonlinear) transformation  $\mathbf{T}: R^n \rightarrow R^n$ , defining  $\mathbf{x} = \mathbf{T}(\mathbf{y})$  exists by which a nonlinear state equation of the form in Eqn. (2.24) (although Bishop explicitly defines his stochastic differential and integral equations "in the sense of Statonovich," rather than Itô) can be transformed into an approximate observer (state) canonical form:

$$\dot{\mathbf{y}}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{b}(\mathbf{y}(t)) + \mathbf{w}(t) \quad (\text{C.21})$$

with *continuous-time* measurements given by:

$$\mathbf{z}(t) = \mathbf{C}\mathbf{y}(t) + \mathbf{v}(t) \quad (\text{C.22})$$

(where  $\mathbf{y}$  is a vector in the transformed space,  $\mathbf{A}$  and  $\mathbf{C}$  are block diagonal matrices,  $\mathbf{b}$  is a nonlinear transformation acting on  $\mathbf{y}(t)$ , and other variables are as defined earlier, except that the measurement noise  $\mathbf{v}(t)$  parameters now conform to a continuous-time measurement description), then a stable estimator of the form in Eqns. (2.16) and (2.18) (i.e., working directly in the untransformed or original vector space) can be defined without explicit computation of the transformation  $\mathbf{T}$ . Bishop finds that this estimator provides improved tracking performance over an extended Kalman filter using the same (planar turn) target model, and proves that his geometric nonlinear filter is locally asymptotically stable for deterministic inputs, and that the second non-central moment of the estimate error probability density is bounded for stochastic inputs.

The Bishop development does not address the issue of the sensitivity of operating this continuous-time filter on a digital computer with discrete-time measurements – the usual environment for modern estimators. In general, operating a continuous-time filter as a discrete-time process, rather than starting with a discrete-time model and *then* designing the estimator, can lead to performance problems [153:43,172].

## *C.6 Equations and Additional Results for Optimal Smoothing*

*C.6.1 Overview.* Implementation of the target recognition approach in Chapter V requires an estimate of target acceleration based on kinematic tracking information only. For this purpose, the author implemented an optimal smoothing routine. This proposal to use optimal smoothing in a *tactical* application is believed to be relatively new – the only published references known to make a similar proposal are by Mahalanabis *et al.* [148, 149]. Note that optimal smoothing for reentry vehicle state estimation in *strategic* defense and space operations is believed to be relatively common (e.g., see [51]).

The implementation of a fixed lag smoother actually starts with the operation of a fixed point smoother for as many measurement intervals as required to achieve the desired lag between the current time and the first time for which the fixed lag smoother state estimate is required. Accordingly, we present the equations for both forms of smoother, as implemented for this effort. All equations are taken from [155:1–18], which in turn refers to [159] and [160], among others. We also consider details of smoother implementation using output from the Kalman filter simulator MSOFE [167]. All of the information required to obtain these smoother estimates is obtained from the concurrently-operating conventional 9-state Kalman filter with Gauss-Markov acceleration (i.e., Singer model [202], as described in Sects. 5.5.2 and 2.3.2.1).

The smoother equations given here provide an optimal estimate for the case of targets described by linear state models, driven by (heuristically) continuous white Gaussian noise, having measurements that are a linear function of the states, corrupted by additive discrete white Gaussian noise. We refer to these as the classic “linear, white, Gaussian” filter and measurement model assumptions, as discussed in Sect. 2.3.1.1. None of these requirements is ever *actually* satisfied in practice. In particular, *real* systems are infinite



dimensional, while *feasible* estimators are inherently finite dimensional. Thus, our models are purposefully of reduced order, and we seek a level of modelling fidelity that balances feasibility with accuracy.

In our case, the (truth model) aircraft targets were modelled as moving instantaneously from linear flight to constant turn rate (i.e., constant speed, constant non-zero specific force normal to the velocity vector) turns lasting 5-10 seconds, and then returning to linear flight. Thus, a Gauss-Markov acceleration filter model is hardly a perfect model. However, the filter dynamics are linear, and, since measurements are given as position and one (doppler) velocity component in a sensor frame, the measurement model is linear with respect to the states (although the measurement matrix  $\mathbf{H}$  changes with target position relative to the sensor, or effectively, with time). In any case, the “optimal” smoother performed quite well.

However, since smoother-derived acceleration states were still rather more noisy in our research than desired for defining target acceleration, target acceleration was ultimately estimated by fitting smoother-derived *position* as a function of time to second-degree polynomial curves, and differentiating twice to find acceleration. This process is also discussed.

Finally, additional results are presented and discussed. Plots were generated using Matlab [151].

*C.6.2 The Fixed Point Smoother.* The fixed point smoother gives the optimal state estimate  $\hat{\mathbf{x}}(t_i | t_j)$  for some (discrete) time  $t_i$ , conditioned on measurements and corresponding Kalman filter outputs through some equal or later time  $t_j$ , where  $\hat{\mathbf{x}}(t_i | t_i)$  is simply  $\hat{\mathbf{x}}(t_i^+)$  from the concurrently running Kalman filter. This smoothed estimate is found using the following equations, for  $t_j \geq t_{i+1}$ :

$$\hat{\mathbf{x}}(t_i | t_j) = \hat{\mathbf{x}}(t_i | t_{j-1}) + \mathbf{W}(t_j)[\hat{\mathbf{x}}(t_j^+) - \hat{\mathbf{x}}(t_j^-)] \quad (\text{C.23})$$

where:

$$\mathbf{W}(t_j) \stackrel{\text{def}}{=} \prod_{k=i}^{j-1} \mathbf{A}(t_k) = \mathbf{W}(t_{j-1})\mathbf{A}(t_{j-1}) \quad (\text{C.24})$$

and:

$$\mathbf{A}(t_k) \stackrel{\text{def}}{=} \mathbf{P}(t_k^+) \Phi^T(t_{k+1}, t_k) \mathbf{P}^{-1}(t_{k+1}^-) \quad (\text{C.25})$$

The terms in these equations are found as follows:

$\hat{\mathbf{x}}(t_j^+)$  = the filter state estimate following update at time  $t_j$ .

$\hat{\mathbf{x}}(t_j^-)$  = the filter state estimate prior to update at time  $t_j$ .

$\mathbf{P}(t_k^+)$  = the filter covariance estimate following update at time  $t_k$ .

$\mathbf{P}(t_{k+1}^-)$  = the filter covariance estimate prior to update at time  $t_{k+1}$ .

$\Phi^T(t_{k+1}, t_k)$  = the transpose of the classical state transition matrix  $\Phi(t_{k+1}, t_k)$  for propagation of primal state variables from time  $t_k$  to  $t_{k+1}$  (i.e.,  $\Phi^T(t_{k+1}, t_k)$  is the state transition matrix for propagating adjoint quantities *backward* in time).

Note that  $\mathbf{W}(t_i) = \mathbf{I}_{n \times n}$ , an identity matrix of order  $n$  (the dimension of  $\mathbf{x}$ ).

If the classical "linear, white, Gaussian" modelling assumptions apply (as in Sect. 2.3.1.1), the covariance  $\mathbf{P}(t_i | t_j)$  of the estimate  $\hat{\mathbf{x}}(t_i | t_j)$  (alternatively, the covariance of the error in this estimate) is given by:

$$\begin{aligned} \mathbf{P}(t_i | t_j) &= \mathbf{P}(t_i | t_{j-1}) + \mathbf{W}(t_j)[\mathbf{P}(t_j^+) - \mathbf{P}(t_j^-)]\mathbf{W}^T(t_j) \\ &= \mathbf{P}(t_i | t_{j-1}) - \mathbf{W}(t_j)\mathbf{K}(t_j)\mathbf{H}(t_j)\mathbf{P}(t_j^-)\mathbf{W}^T(t_j) \end{aligned} \quad (\text{C.26})$$

where all terms are defined above or in Sect. 2.3.1.1, and  $\mathbf{P}(t_i | t_i)$  is simply  $\mathbf{P}(t_i^+)$  from the concurrently running Kalman filter. If the classical assumptions do not apply, then Eqn. (C.26) at least provides an estimate of the estimate or error covariance.

*C.6.3 The Fixed Lag Smoother.* The optimal state estimate for an  $N$ -step time lag is given by:

$$\begin{aligned}\hat{\mathbf{x}}(t_{i+1} | t_{i+N+1}) &= \Phi(t_{i+1}, t_i) \hat{\mathbf{x}}(t_i | t_{i+N}) \\ &\quad + \mathbf{C}(t_{i+N+1}) \mathbf{K}(t_{i+N+1}) [\mathbf{z}_{i+N+1} - \mathbf{H}(t_{i+N+1}) \hat{\mathbf{x}}(t_{i+N+1}^-)] \\ &\quad + \mathbf{U}(t_{i+1}) [\hat{\mathbf{x}}(t_i | t_{i+N}) - \hat{\mathbf{x}}(t_i^+)]\end{aligned}\quad (\text{C.27})$$

where the  $n$ -by- $n$  "gain" matrix  $\mathbf{C}(t_{i+N+1})$  is given by

$$\mathbf{C}(t_{i+N+1}) = \prod_{k=i+1}^{i+N} \mathbf{A}(t_k) = \mathbf{A}^{-1}(t_i) \mathbf{C}(t_{i+N}) \mathbf{A}(t_{i+N}) \quad (\text{C.28})$$

with  $\mathbf{A}(t_k)$  defined in Eqn. (C.25); and the  $n$ -by- $n$  matrix  $\mathbf{U}(t_{i+1})$  is given by

$$\mathbf{U}(t_{i+1}) = \mathbf{G}_d(t_i) \mathbf{Q}_d(t_i) \mathbf{G}_d^T(t_i) \Phi^T(t_i, t_{i+1}) \mathbf{P}^{-1}(t_i^+) \quad (\text{C.29})$$

Under the usual "linear, white, Gaussian" assumptions, the smoothed estimate has zero-mean error, and the covariance of the estimate and/or error is given by:

$$\begin{aligned}\mathbf{P}(t_{i+1} | t_{i+N+1}) &= \mathbf{P}(t_{i+1}^-) - \mathbf{C}(t_{i+N+1}) \mathbf{K}(t_{i+N+1}) \mathbf{H}(t_{i+N+1}) \mathbf{P}(t_{i+N+1}^-) \mathbf{C}^T(t_{i+N+1}) \\ &\quad - \mathbf{A}^{-1}(t_i) [\mathbf{P}(t_i^+) - \mathbf{P}(t_i | t_{i+N})] \mathbf{A}^{-1}(t_i)^T \\ &= \mathbf{P}(t_{i+1}^-) - \mathbf{C}(t_{i+N+1}) [\mathbf{P}(t_{i+N+1}^-) - \mathbf{P}(t_{i+N+1}^+)] \mathbf{C}^T(t_{i+N+1}) \\ &\quad - \mathbf{A}^{-1}(t_i) [\mathbf{P}(t_i^+) - \mathbf{P}(t_i | t_{i+N})] \mathbf{A}^{-1}(t_i)^T\end{aligned}\quad (\text{C.30})$$

The fixed lag smoother is initialized using a state estimate  $\hat{\mathbf{x}}(t_0 | t_N)$  and covariance estimate  $\mathbf{P}(t_0 | t_N)$  found by iterating the fixed point smoother for fixed point  $t_0$  a total of  $N$  times. These values become  $\hat{\mathbf{x}}(t_i | t_{i+N})$  in Eqn. (C.27) and  $\mathbf{P}(t_i | t_{i+N})$  in Eqn. (C.30), respectively.

**C.6.4 Optimal Smoothing with MSOFE Output.** In the Kalman filter simulator MSOFE [167], all outputs are controlled from subroutine OUT. For smoother implementation using MSOFE outputs, the pre-measurement update quantities  $\hat{\mathbf{x}}(t_j^-)$  and  $\mathbf{P}(t_j^-)$  are available after subroutine USROUT is called from subroutine OUT with calling argument IOUT = 3. Post-measurement quantities  $\hat{\mathbf{x}}(t_j^+)$  and  $\mathbf{P}(t_j^+)$  are available after subroutine USROUT is called with IOUT = 5.

Since MSOFE performs sequential scalar updates, the quantities  $\mathbf{K}(t_j)$  and  $\mathbf{H}(t_j)$  are made available vector by vector at each scalar update after subroutine USROUT is called with IOUT = 4 ( $m$  events per set of scalar measurements). These quantities are saved at each IOUT = 4 call and output as matrices with post-measurement state updates at each IOUT = 5 event. Note that  $\mathbf{K}(t_j)$  is provided column-by-column, while  $\mathbf{H}(t_j)$  is provided row-by-row. Similarly, the expected measurement vector  $\mathbf{H}(t_j)\hat{\mathbf{x}}(t_j^-)$  and corresponding noise-corrupted measurement are made available scalar by scalar, and output as vectors at each IOUT = 5 event.

It is important to understand, however, that the Kalman filter gain  $\mathbf{K}(t_j)$  and expected measurement values  $\mathbf{H}(t_j)\hat{\mathbf{x}}(t_j^-)$  generated in this way from successive *scalar* measurement updates in MSOFE are not the same as those which would be generated by a classical *vector* measurement update filter. As discussed in Sect. 4.3, where successive measurements provide non-independent information about the same states, or measurement noise is not independent between measurements, we may expect that this MSOFE-provided sequence of expected measurement values and Kalman filter gains reflect information from previous scalar updates. In MSOFE, the classical Kalman gain matrix and expected measurement value vector are not generally available prior to the sequence of measurement updates.

For example, in our case, each of the three target position measurements in general provides some information about target position along each of the three inertial axes. However, since the “sensor frame” axes are orthogonal to one another, and measurement noise is considered independent on each axis, the position measurements are in fact independent from one another with respect to the *information* each provides about any one state. Therefore, we do not expect that one sensor frame scalar position measurement update

will change the expected measurement value for a subsequent scalar position measurement in the same measurement set. On the other hand, the doppler velocity expected measurement is *not* independent from the prior measurements: in particular, it is affected by the prior measurement of range, or position along the sensor "boresight" axis – thus expected doppler measurements before and after the scalar position measurements will differ.

The smoother equations given here and in [154:13-14] apply to the vector update case. With careful manipulation and consideration, however, these equations can be made optimal as well for information available from MSOFE, as taken from sequential scalar updates. First of all, the reader should note that the quantity  $\mathbf{K}(t_{i+N+1})[\mathbf{z}_{i+N+1} - \mathbf{H}(t_{i+N+1})\hat{\mathbf{x}}(t_{i+N+1}^-)]$  from Eqn. (C.27), or  $\hat{\mathbf{x}}(t_{i+N+1}^+) - \hat{\mathbf{x}}(t_{i+N+1}^-)$ , is exactly the *same* vector quantity whether the Kalman gain matrix  $\mathbf{K}(t_{i+N+1})$  and the residual vector  $[\mathbf{z}(t_{i+N+1}) - \mathbf{H}(t_{i+N+1})\hat{\mathbf{x}}(t_{i+N+1}^-)]$  are computed before the measurement update, or assembled afterward using Kalman gain vectors and scalar residuals from successive scalar updates. Similarly, although the quantity  $\mathbf{K}(t_{i+N+1})\mathbf{H}(t_{i+N+1})\mathbf{P}(t_{i+N+1}^-)$  will be computed incorrectly in Eqn. (C.30) if  $\mathbf{K}$  is assembled from MSOFE Kalman gain vectors, the correct quantity is available in any case as  $\mathbf{P}(t_{i+N+1}^-) - \mathbf{P}(t_{i+N+1}^+)$ , computed from filter covariance values (respectively) before any and after all measurements.

Note, however, that further complications are to be expected regarding the relationship between scalar and vector measurement updates if feedback control, sensor failures, or nonlinear processes come into play from scalar update to update. If desired, the classical  $\mathbf{K}(t_j)$  matrix can be reconstructed using pre-measurement information and Eqn. (2.11), and expected measurement values can likewise be found prior to any scalar updates. For the fixed point smoother, the need to reconstruct the matrix  $\mathbf{K}(t_j)$  in this way is avoided in any case by using the first equality in Eqn. (C.26).

To limit complication in the MSOFE code, the author's smoother uses MSOFE scalar-update-generated Kalman gains and residuals, subject to the considerations noted above, subsequently assembled into vector-matrix form. These MSOFE-generated quantities were written to a file "SMOODAT" with a format similar to that of the MSOFE output file "CTOM" (plot output for continuous variables), which is processed by program MPLOT (a plot post-processor for MSOFE output). The Smoother/Recognizer

program, then, was a derivation of MPlot which routed input data from "SMOODAT" into smoother subroutines rather than plot processing subroutines.

The only problem encountered in operating the fixed point and fixed lag smoothers was an instability problem in the fixed lag smoother after 3-4 seconds in single precision, or 15 seconds in double precision. At least part of this instability is believed due to sub-optimal covariance matrix inversions associated with Eqn. (C.25), in which old information is "backed out" of the smoother. The smoother was first implemented in single precision, which implies a 4-byte word length on SPARC workstations. The rapid development of instability in this mode was investigated, and it was found that single precision matrix inversion, followed by multiplication of the original matrix with its inverse, resulted in off-diagonal terms as large as  $10^{-3}$  - i.e., significantly non-zero. It was clear that 30-40 iterations of this process over 3-4 seconds could introduce significant numerical error into the computations.

Double precision inversion and double precision multiplication were required to keep off-diagonal terms smaller than  $10^{-8}$ . Running the smoother with double precision operations allowed the algorithm to run for lengths of time sufficient to prove the concept, but, even in double precision, the fixed-lag smoother will become unstable after 15 or so seconds.

This instability is but a minor concern for at least four reasons. First, in an actual on-line implementation, most smoother parameters could be kept as fixed gains, and operations like Eqn. (C.25) would not be implemented on-line. Second, if it were necessary to do so, the smoother can be "freshened" by restarting the recursive computation of Eqn. (C.25) with stored values. Third, one may choose to implement a "fully factored" or numerically stable form of smoother [31]. Finally, as noted, the time period prior to instability was adequate for our purposes in any case.

*C.6.5 Curve Fitting.* As discussed in Sect. 5.6, the smoother-derived target acceleration estimate *per se* was much closer to the true target acceleration than the filter-derived acceleration estimate, but smoother-derived acceleration was still rather more noisy than desired for algorithm functioning. It must be emphasized that the author's algorithm

is designed to identify target accelerations relative to the body frame that remain reasonably constant within defined bounds (nominally  $\pm 5$  feet per second squared over an 0.8 second period) – indicative of a steady state condition in the target maneuver. The identification of this condition and a corresponding nominal acceleration level define the start of the recognition process – for this reason, a smooth, if not perfectly accurate, acceleration estimate is essential. Errors in the accuracy of the acceleration estimate are overcome by the dynamic programming “motion warping” process.

The final acceleration estimation method selected was to fit smoother-derived position to a second degree (quadratic) polynomial curve, and differentiate twice to find acceleration. This curve fitting was accomplished with the use of the IMSL utility “RCURVE” [114], which provides least-squares polynomial curve fit parameters for a curve of given degree to fit an array of points. In this application, the smoother-derived position along each inertial (filter) axis is maintained as a set of 43 values at 0.1 sec. intervals, covering an elapsed time of 4.2 seconds, and the acceleration is taken at the midpoint of the curve. Thus, typically, a total of 4.1 seconds is required before the curve fit-derived acceleration estimate is available for some given (real) time. That is, 4.1 sec. = 2.0 sec. smoother lag + 2.1 sec. (= 0.5 [4.2 sec.]) curve fit delay.

It should be noted that the use of the IMSL utility “RCURVE” is a simple, but inefficient way to implement a least squares curve fit. Had this curve fit been performed internally to the author’s code, a recursive update to the curve fit parameters would have been possible, with low computational load.

Other approaches for acceleration state estimation by curve fitting to position measurements are discussed in [50] and [74]. The author does not recommend acceleration state estimation by curve fitting to Kalman filter-derived position measurements *per se*, for reasons that will be clear in the following section.

*C.6.6 Results and Discussion.* In this section, we present smoother results from four different tracking scenarios. Some results shown in Chapter V were taken from a fifth scenario – all five scenarios are presented in Fig. C.1. With reference to the results in Chapter V, the first scenario corresponds to Figs. 5.26, 5.27, and 5.29 through 5.31,

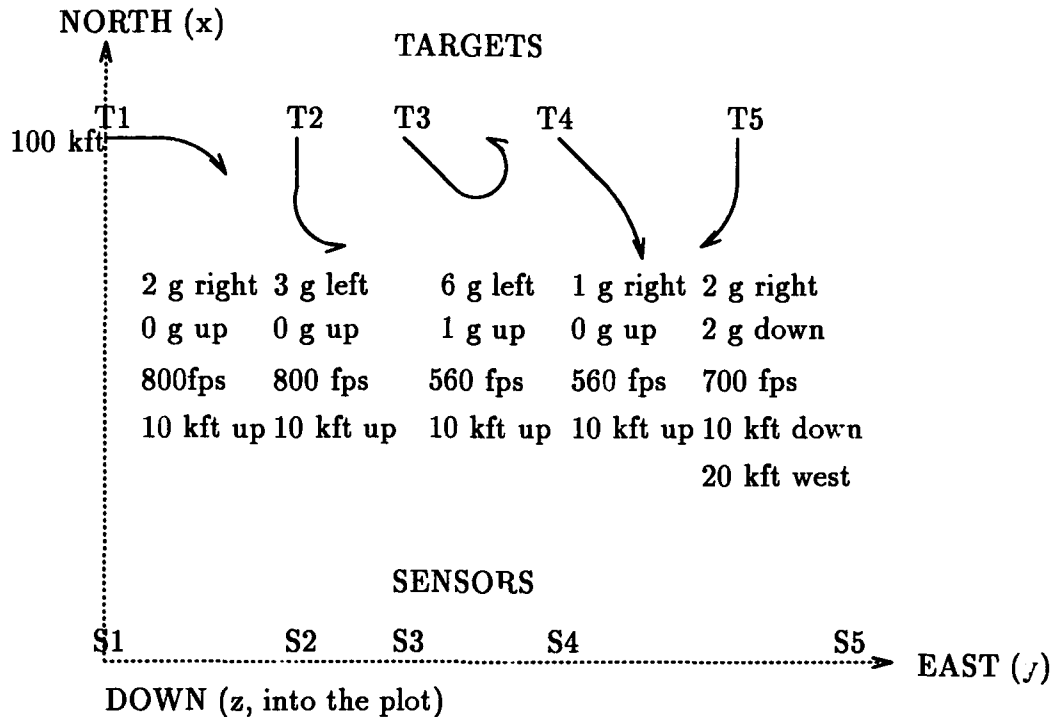


Figure C.1. Target Trajectories for Smoother Discussion

while the second scenario corresponds to Figs. 5.28 and 5.32. These results concentrate on acceleration state estimation, since that is by far the most critical variable for our purposes. The plots show mean error in state estimates by the (1) Kalman filter, (2) smoother, and, where required, (3) least squares quadratic curve fit to smoothed position, respectively, for Monte Carlo sets of 20 runs with the parameters given in Chapter V. Also shown are the true mean error and mean  $\pm$  one standard deviation bounds for the filter and smoother, as well as the filter and smoother-predicted error standard deviations (filter and smoother-predicted mean error are zero by definition). Standard deviation curves for the true error in the curve fit estimate are not shown to maintain clarity, but are on the order of or slightly smaller than the corresponding standard deviations for the smoother true error.

In each case, target accelerations last for eight seconds, starting at 2.0 or 3.0 seconds, depending on the scenario (maneuver onset time is clear from the plots). In all but the



last case, the Kalman filter track begins at 0.0 seconds, and the smoothed estimate starts at 2.0 seconds and runs until 11.9 seconds (using *a posteriori* information from 4.0 to 13.9 seconds, respectively – a two-second time lag). The last case uses a one-second time lag, and the unique timing issues for that case will be noted below. The difference in start time for the maneuvers is simply to show that the smoother performance is robust with respect to changes in start time. Figures C.2 and C.3 show the error in east (inertial y axis) target acceleration and position, respectively, for Scenario 2. The position error plot does not show a mean curve fit position error since this plot follows the mean smoother error curve exactly.

Note that the positive position estimate error indicates that the smoothed position estimate *leads* the true position somewhat, while the filter position lags significantly. The key point, alluded to above, regarding this filter position estimate lag is that curve fitting to *filter* position estimates, followed by twofold differentiation to find acceleration, did *not* yield adequate acceleration estimates – the typical position estimate “lag” as seen here translates into a low acceleration estimate after curve fitting and differentiation. The relatively small position lead error from the smoother has an unnoticeable effect on the acceleration estimate.

Also, note that just before the commanded acceleration ceases at 11 seconds, the smoother position estimate starts to lag the true position. This is due to the fact that the smoother is “aware” that the acceleration will soon cease, and has begun to react appropriately.

Figures C.4, C.5 and C.6 show the error in east (inertial y axis) target acceleration and position, and downward (inertial z axis) target acceleration, respectively, for Scenario 3. This scenario is provided to show that smoothing provides considerable robustness to choices of target acceleration model parameters. Recall from Chapter V that this filter is tuned for a (“benign”) target acceleration standard deviation of one g (32 feet per second squared) – clearly, the filter cannot follow the 6 g target in Scenario 3 at all well, but the smoother reduces state estimation errors dramatically. This performance is achieved even though this target, due to its relatively low speed and high g turn, actually reverses its direction over the course of the maneuver. Thus, it appears that we can use “benign” filter

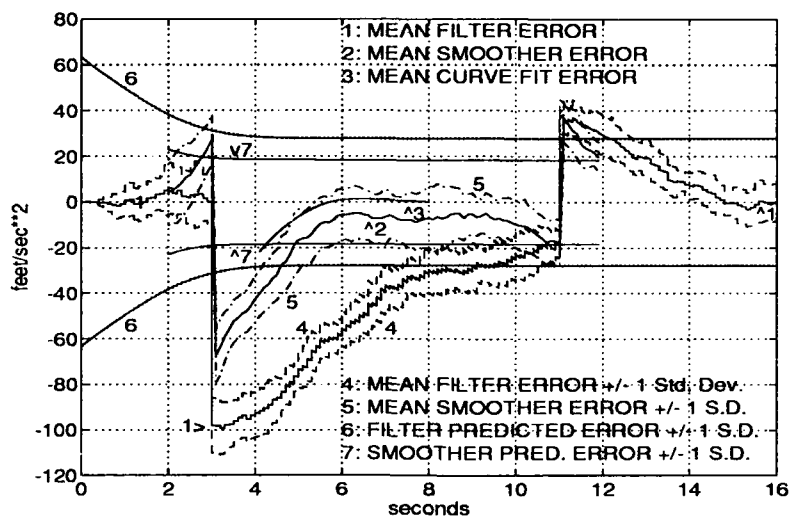


Figure C.2. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 2

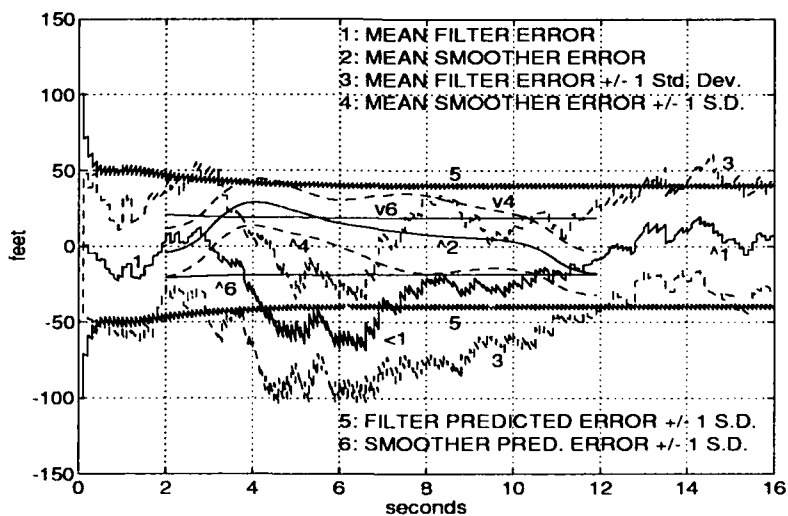


Figure C.3. Error in Target Position Estimates – Inertial Y Axis, Scenario 2

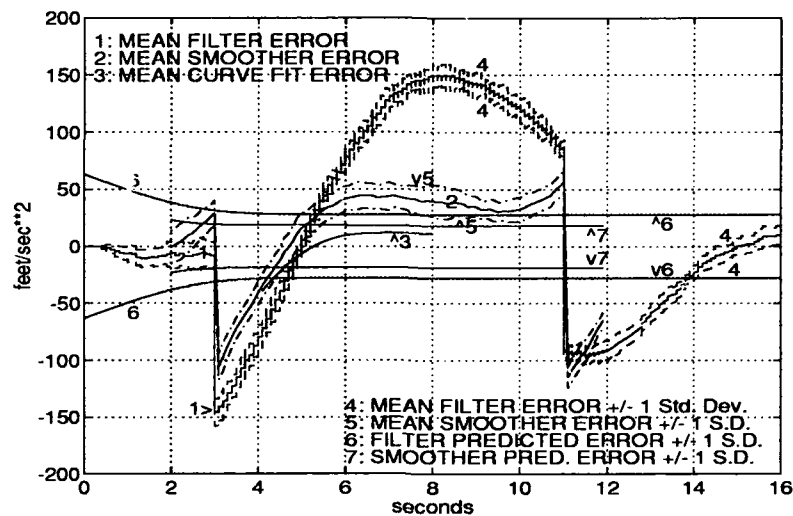


Figure C.4. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 3

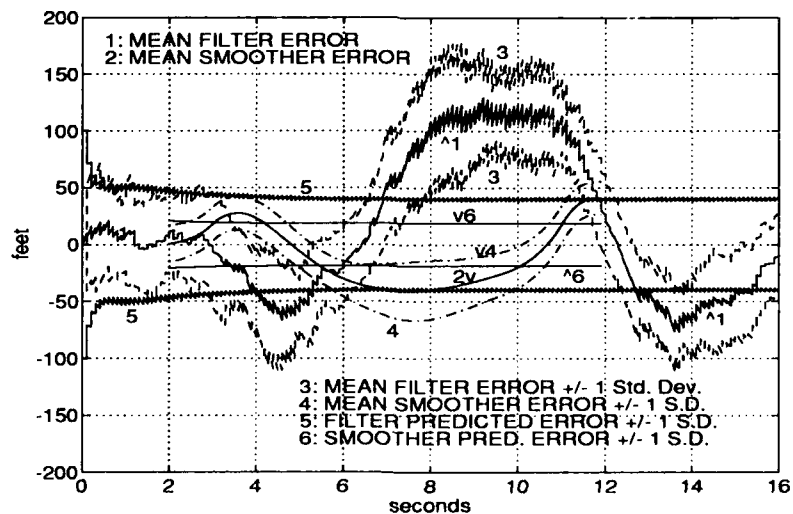


Figure C.5. Error in Target Position Estimates – Inertial Y Axis, Scenario 3

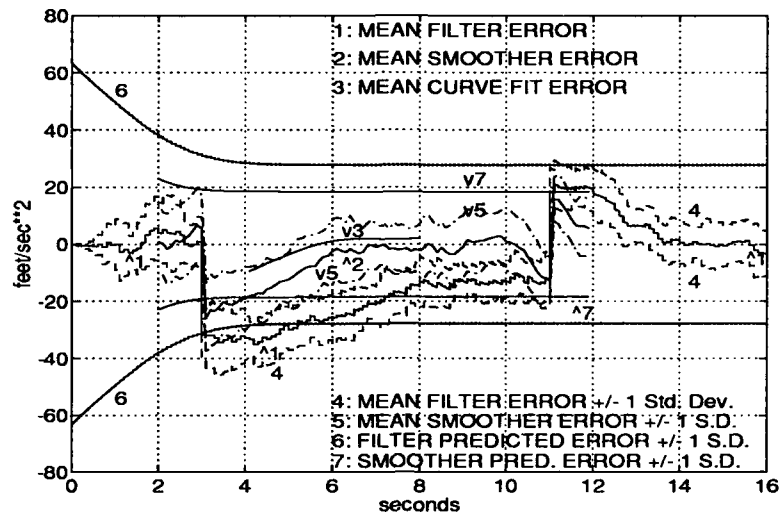


Figure C.6. Error in Target Acceleration Estimates – Inertial Z Axis, Scenario 3

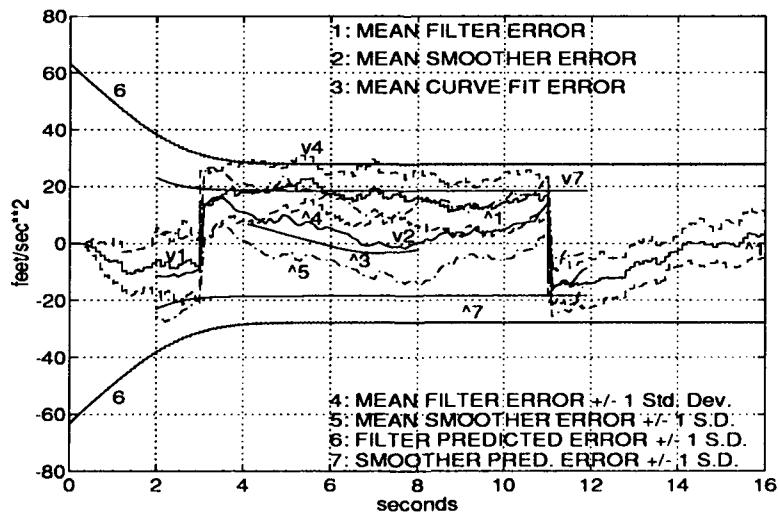


Figure C.7. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 4

tuning parameters and thereby “damp out” trajectory estimate error due to measurement noise, but preserve low estimation errors for highly dynamic targets by smoothing.

Figures C.7 and C.8 show the error in east (inertial y axis) and downward (inertial z axis) target acceleration, respectively, for Scenario 4. Fig. C.7 shows that improvement in the state estimate is gained even for benign maneuvers, while Fig. C.8 demonstrates that optimal smoothing does not provide improvement in the absence of an unmodelled external driving force (an unmodelled “Q”), as discussed in [154:13–14].

Figures C.9 and C.10 show the error in east (inertial y axis) and downward (inertial z axis) target acceleration, respectively, for Scenario 5. Results here are consistent with previous cases, for this somewhat different trajectory.

Finally, to observe the effect of a reduced lag time, Fig. C.11 repeats the scenario of Fig. C.9 with a one-second fixed lag time, instead of two seconds. Note the considerable degradation in the accuracy of the smoother-derived acceleration estimate. Some curves are not labeled due to the close spacing, but the general effect of this time reduction is clear. The curve fit-derived estimate is not substantially degraded in the mean, however – a more dynamic turn would have produced more error in this estimate. Note that the smoother estimate now lasts for an additional second, since the smoother cutoff is referenced to the same real time point as in the previous cases, and the smoother now runs until one second prior to that time, rather than two seconds prior, as in the previous cases.

### *C.7 Equations for Direction Cosine Matrix (DCM) Generation*

The purpose of this section is to provide the reader with simple procedures for generating direction cosine matrices (DCM) for tracking problems. We seek to avoid, to the extent that we can, the laborious process of defining Euler angles.

*C.7.1 DCM for Sensor-to-Inertial Frame Transformation.* The following discussion applies equally to the transformations (1) from the *true* line-of-sight or sensor frame to the inertial frame, and the inverse transformation, or (2) from the *filter-predicted* line-of-sight or sensor frame to the inertial frame, and its inverse. The only difference is that in transformation (1), the true target location is used, while in transformation (2), the latest

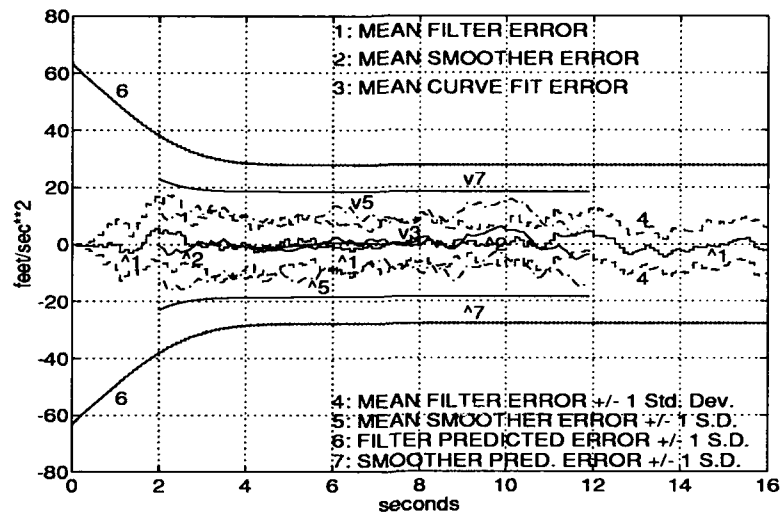


Figure C.8. Error in Target Acceleration Estimates – Inertial Z Axis, Scenario 4

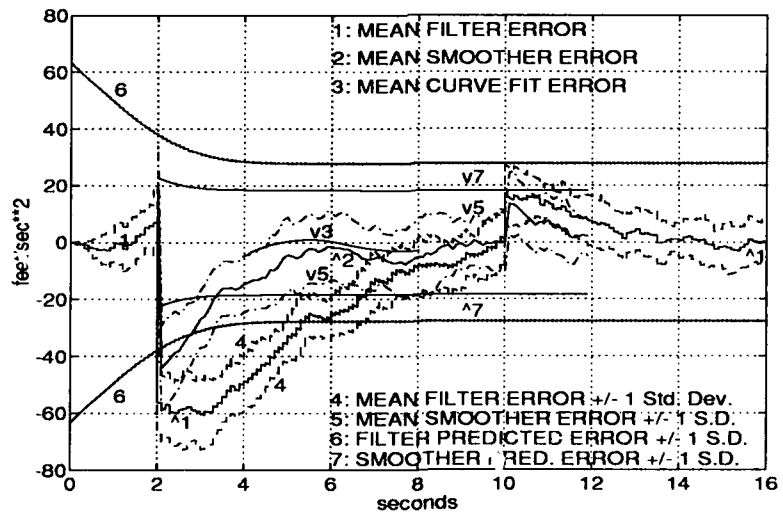


Figure C.9. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 5

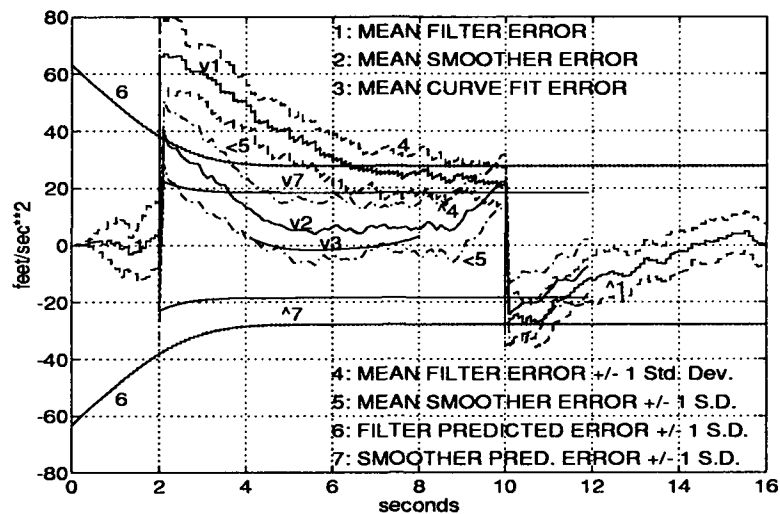


Figure C.10. Error in Target Acceleration Estimates – Inertial Z Axis, Scenario 5

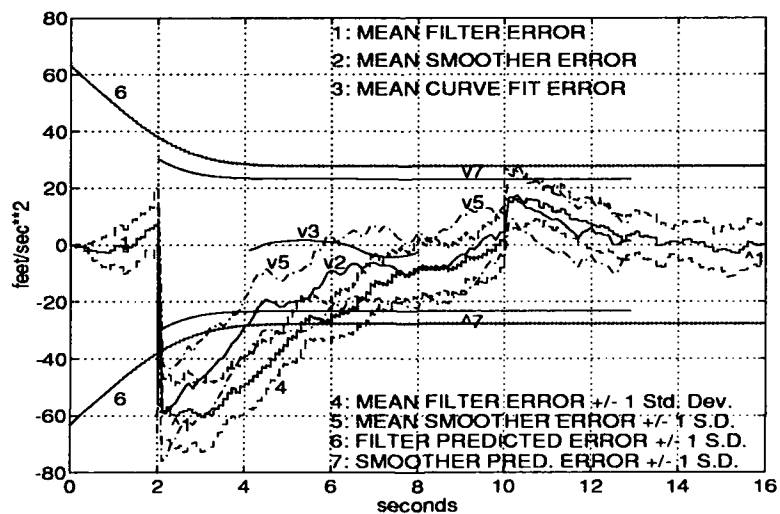


Figure C.11. Error in Target Acceleration Estimates – Inertial Y Axis, Scenario 5 (One Second Fixed Lag)

best filter estimate of target location is used. The first set of transformations, not actually available to the tracking algorithm, is used to define "truth" measurement values, including noise realizations. The second set of transformations is used to define the Kalman filter measurement matrix  $\mathbf{H}$ , expected measurement values, and so on – these transformations are of course available to the filter. Note that this discussion assumes perfect knowledge of sensor or "ownship" inertial position and orientation with respect to the local level. If required, one can further define true and filter-assumed inertial frames (see appropriate Fortran code in [143]).

Obtaining the direction cosine matrix for sensor-to-inertial frame transformation starts by defining the sensor-to-target or *boresight* vector  $\mathbf{i}_B$  in inertial frame coordinates. Dividing this vector by its length (range), we obtain the sensor-to-target unit vector,  $\mathbf{i}_B$ .

Now, take the vector cross product of the local downward or inertial unit vector  $\mathbf{i}_z$  ( $[0, 0, 1]^T$ , in inertial coordinates) with the sensor-to-target unit vector ( $\mathbf{i}_z \times \mathbf{i}_B$ ). The resulting "elevation axis" vector is generally of non-unit length, and, by definition of the cross product, normal to the two vectors crossed to produce it. Thus, it is parallel to the local horizontal. Now divide the elevation axis vector by its length to normalize it, producing the elevation axis unit vector  $\mathbf{i}_E$  in inertial frame coordinates.

Next, cross the sensor-to-target unit vector,  $\mathbf{i}_B$  into the elevation axis unit vector  $\mathbf{i}_E$  ( $\mathbf{i}_B \times \mathbf{i}_E$ ) to produce the azimuth axis unit vector  $\mathbf{i}_A$  in inertial frame coordinates. Since the two unit vectors crossed to produce  $\mathbf{i}_A$  were normal or perpendicular to each other, their cross product is automatically of unit length also, and does not require subsequent normalization. The sensor frame is now defined by  $\mathbf{i}_B$ ,  $\mathbf{i}_E$ , and  $\mathbf{i}_A$ , as shown schematically in Fig. 4.1.

Each of  $\mathbf{i}_B$ ,  $\mathbf{i}_E$ , and  $\mathbf{i}_A$  is a three-element column vector of inertial frame coordinates. Arranging these three vectors vertically side-by-side as a three by three matrix, we immediately have the direction cosine matrix  $\mathbf{C}_{i_s}^i$  for converting from the sensor or line-of-sight (subscript  $s$ ) frame to the inertial (superscript  $i$ ) frame. That is, multiplying  $\mathbf{C}_{i_s}^i$  times a vector coordinatized in the sensor frame (vector element order: boresight axis component, elevation axis component, azimuth axis component) yields a vector coordinatized in



the inertial frame. Finally, since  $C_{i_s}^i$  is a *unitary* matrix, its inverse  $C_{i_s}^{i'}$  is equal to its transpose.

*C.7.2 DCM for Target Lift-to-Inertial Frame Transformation.* In Chapter V, we discussed definitions for the target velocity and target body coordinate frames. The purpose of this section is to expand upon that discussion and consider some details of the author's implementation and use of coordinate frames.

In Sect. 5.5.3.1, we found a representation for the velocity frame and the DCM transformation  $C_i^v$  from inertial to velocity frames using the classical (Euler) angles of heading and flight path angle. Following the discussion on the sensor frame in the previous section, the reader may now note that the velocity frame is defined relative to the velocity unit vector exactly as the sensor frame is defined relative to the boresight vector. This fact presents a convenient alternative to Euler angles for obtaining  $C_i^v$  – i.e., through vector cross products as in the previous section. The result is equivalent element for element to that found using Euler angles as shown in Eqn. (5.4).

As a useful intermediate between the velocity and body coordinate frames, the author's simulations occasionally make use of a "lift" frame. This lift frame (denoted by subscript  $L$ ) is essentially identical to the "wind" frame (denoted by subscript  $w$ ) as defined by Etkin [79], distinguished from the inertial frame by heading angle  $\eta$ , flight path angle  $\gamma$ , and a roll angle  $\rho$ , in that order. The author desires to distinguish between Etkin's wind frame and the lift frame, however, since the last Euler angle (roll) taking the inertial frame into the Etkin's wind frame is defined by a vehicle plane of symmetry, while in our case, this angle is defined by the direction of the aerodynamic normal load or lift vector. In fact, these definitions are equivalent in most cases, but the possibility of a difference should be kept in mind.

Definition of the DCM for target lift-to-inertial frame transformation starts by defining  $C_i^v$  using either Euler angles or vector cross products. Depending upon the application, target state values may be actual (true) or estimated quantities, and simulation code must ensure that appropriate values are used.

Next, define the target acceleration normal to and along the velocity vector. The desired inertial acceleration normal to the velocity vector can be isolated into velocity frame components (1)  $a_{y_v}$  parallel to the local horizontal, and orthogonal to the velocity vector, and (2)  $a_{z_v}$ , generally downward, mutually orthogonal to the velocity vector and  $a_{y_v}$ . The specific force normal to the velocity vector required to achieve this acceleration is assumed due to aerodynamic lift, and can be isolated into components (1)  $a_{y_v}$ , (2)  $a_{z_v}$ , and (3)  $g \cos(\gamma)$ , parallel to  $a_{z_v}$  and counteracting the component of gravity along the  $z_v$  axis. The angle  $\rho$  is then calculated as shown in Fig. 5.24. Note that lift is positive in the direction opposite to the lift frame  $z_L$  vector, which points generally downward.

The lift frame and the velocity frame are distinguished only by the roll angle  $\rho$  about the velocity vector. Thus, we obtain the DCM for transformations from the inertial to the lift frame by premultiplying  $\mathbf{C}_i^v$  by the direction cosine matrix  $\mathbf{C}_v^L$ , which is found as:

$$\mathbf{C}_v^L = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & \cos(\rho) & -\sin(\rho) \\ 0 & \sin(\rho) & \cos(\rho) \end{bmatrix} \quad (\text{C.31})$$

The usual assumptions for aircraft wind and body frames call for the transformation from wind frame to body frame to proceed by a two-part rotation: first about the  $z_w$  axis by a sideslip angle  $\beta$ , followed by an angle-of-attack rotation  $\alpha$  about the (now rotated)  $y_w$  axis (nominally, through the wings), as shown in Fig. 5.23 and [79]. As noted in Sect. 5.5.3.1, however, the nominal assumption in this research calls for a zero sideslip angle. For that reason, the nominal target body frame is generally found by rotating the lift frame directly by an angle of attack  $\alpha$  about the  $y_L$  axis (the lift frame axis mutually orthogonal to the lift and velocity vectors – generally, through the wings, as for the wind frame  $y_w$  axis [79]).

For the development in Chapter IV, it was desirable also to model sideslip angle, but transformations from the lift frame to the body frame involving this angle were modelled as proceeding in the order of angle of attack first, then sideslip. This was done because angle of attack is the major variable in force calculations, and the author did not desire

to recompute angle of attack following changes in sideslip angle. In any case, for *small* angular displacements, effects from these two rotations about orthogonal axes may be *approximated* as independent of order, as motivated in [187:2.17].

### *Vita*

Lieutenant Colonel Edmund W. Libby was born in 1953 in Ann Arbor, MI, and grew up in Studio City, CA. He received the Bachelor of Science degree with honors in Chemical Engineering from the University of California at Berkeley in 1975, and accepted a Regular Commission in the Ordnance Corps of the United States Army. Prior to entering on active duty, he worked as a civilian technician on the U.S. Navy's 5" Cannon-Launched Guided Projectile (CLGP) Program at the Naval Weapons Center, China Lake, CA. His Army assignments include nearly six years of service in Germany as a missile/armament/munitions officer, command of a foreign materiel maintenance company, and duty as a systems engineer in a major program of the Strategic Defense Initiative (SDI). He received the Master of Science degree with distinction in Guidance and Control from the U.S. Air Force Institute of Technology (AFIT) in 1983, and is a member of Tau Beta Pi and Eta Kappa Nu. He is blessed to be married to the former Patricia Ann McCarthy of Danville, VA, and to have three wonderful children.

Permanent address: 115 Shadowwood Ct.  
Danville, VA 24540

### Bibliography

1. Richard A. Altes. "Target Position Estimation in Radar and Sonar, and Generalized Ambiguity Analysis for Maximum Likelihood Parameter Estimation," *Proceedings of the IEEE*, 67(6):920-930 (June 1979).
2. Richard A. Altes. "Radar/Sonar Acceleration Estimation With Linear-Period Modulated Waveforms," *IEEE Trans. AES*, 26(6):914-924 (November 1990).
3. Amir A. Amini, Terry E. Weymouth, and Ramesh C. Jain. "Using Dynamic Programming for Solving Variational Problems in Vision," *IEEE Trans. Patt. Analy. and Mach. Intel.*, 12(9):855-866 (September 1990).
4. Dominick Andrisani, Eung Tai Kim, John Schierman, and Frank P. Kuhl. "A Nonlinear Helicopter Tracker Using Attitude Measurements," *IEEE Trans. Aerospace and Electronic Systems*, 27(1):40-47 (January 1991).
5. Dominick Andrisani, Frank P. Kuhl, and Daniel Gleason. "A Nonlinear Tracker Using Attitude Measurements," *IEEE Trans. Aerospace and Electronic Systems*, 22(5):533-539 (September 1986).
6. Thomas M. Apostol. *Mathematical Analysis*. New York, NY: Addison-Wesley Publishing Co., 1974.
7. James F. Arnold and Yaakov Bar-Shalom *et al.* "Target Parameter Estimation Using Measurements Acquired with a Small Number of Sensors," *IEEE Journal of Oceanic Engineering*, 8(3):163-172 (July 1983).
8. Yaakov Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Advanced Applications*. Norwood, MA: Artech House, 1990.
9. Yaakov Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Advanced Applications, II*. Norwood, MA: Artech House, 1992.
10. Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association*. New York: Academic Press, Inc., 1988.
11. Walter Barnes. "Conversations with the Author Regarding High Range Resolution Radar Target Recognition." (Input from Wright Laboratory Radar Branch), 1993.
12. Yair Barniv. "Dynamic Programming Solution for Detecting Dim Moving Targets," *IEEE Trans. Aerospace and Electronic Systems*, 21(1):144-156 (January 1985).
13. Yair Barniv. "Dynamic Programming Solution for Detecting Dim Moving Targets PART II: Analysis," *IEEE Trans. Aerospace and Electronic Systems*, 23(6):776-788 (November 1987).
14. Anne K. Barrette, L.C. Jamerson, and R.A. Rogers. "Attribute Level Fusion Applied to Type/Class Identification." *Proc. 3d Nat'l Symposium on Sensor Fusion II*, edited by Dr. Rudolf J. Buser (Chairman). 117-134. 1990. (unclassified article in secret volume).
15. David K. Barton. *Modern Radar System Analysis*. Norwood, MA: Artech House, 1988.

16. David K. Barton, Charles E. Cook, and Paul Hamilton, editors. *Radar Evaluation Handbook*. Norwood, MA: Artech House, 1991.
17. M. Basseville, A. Benveniste, A.S. Willsky, and K.C. Chou. "Multiscale Statistical Processing: Stochastic Processes Indexed by Trees." *Proc. of Int'l. Symp. on Math. Theory of Networks and Systems*. June 1989.
18. F. Beckner. "Comments during briefing at Wright-Patterson AFB on High Range Resolution Radar Target Recognition Research." (author is with Cyberdynamics, Inc., Palo Alto, CA, on contract to the Target Recognition Office, Wright Labs), February 1992.
19. F. Beckner. "Phonecon with the author." (Discussing HRRR signature noise modelling and analysis techniques), June 1992.
20. F. Beckner *et al.* *Automatic Radar Target Identification (ARTI) Phase II (U)*. Technical Report WRDC-TR-90-1003, Wright-Patterson AFB, OH: General Dynamics (Pomona) and U.S. Air Force Wright Labs, March 1990.
21. F. Beckner *et al.* *Signature Synthesis*. Technical Report SP-6365-3, 55 Walkers Brook Dr., Reading, MA 01867: TASC, February 1992. (compilation of various contractors' briefings from 12 Feb. 1992 meeting on subject).
22. Thaddeus Beier and Shawn Neely. "Feature-Based Image Metamorphosis," *Computer Graphics*, 26(2):35-42 (July 1992).
23. Richard E. Bellman and Stuart E. Dreyfus. *Applied Dynamic Programming*. Princeton, NJ: Princeton University Press, 1962.
24. T.R. Benedict and G.W. Bordner. "Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations," *IRE Transactions on Automatic Control*, 27-32 (July 1962).
25. R. Berton. "Satellite-Borne Radar," *La Recherche Aerospatiale*, (5):15-28 (May 1990). (published in English and French editions).
26. D. Bestle and M. Zeitz. "Canonical Form Observer Design for Non-Linear Time-Variable Systems," *Int. Jour. Control*, 38(2):419-431 (1983).
27. Bir Bhanu. "Model Based Segmentation of FLIR Images." *Applications of Digital Signal Processing VII, Proc. SPIE504*, edited by Andrew G. Tescher. 10-18. 1984.
28. Bir Bhanu. "Automatic Target Recognition: State of the Art Survey," *IEEE Trans. Aerospace and Electronic Systems*, 22(4):364-379 (July 1986).
29. Bir Bhanu. "Model Based Segmentation of FLIR Images," *IEEE Trans. Aerospace and Electronic Systems*, 26(1):2-10 (January 1990).
30. Bir Bhanu and Terry L. Jones. "Image Understanding Research For Automatic Target Recognition." *Proceedings: Image Understanding Workshop*. 249-254. Defense Advanced Research Projects Agency (DARPA), January 1992.
31. G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. New York, NY: Academic Press, 1977.

32. Robert H. Bishop. *Geometric Nonlinear Filtering Theory with Application to the Maneuvering Aircraft Tracking Problem*. PhD dissertation, Rice University, 1990.
33. Samuel S. Blackman. *Multiple-Target Tracking with Radar Applications*. Norwood, MA: Artech House, 1986.
34. B.Z. Bobrovsky and M. Zakai. "A Lower Bound on the Estimation Error for Markov Processes," *IEEE Trans. Auto. Control* (December 1975).
35. Philip L. Bogler. *Radar Principles with Applications to Tracking Systems*. New York: John Wiley and Sons, 1990.
36. Christopher L. Bowman and Matthew Gross. "Multi-Sensor Multi-Platform Track Association Using Kinematics and Attributes." *Proc. IEEE NAECON*. 204-208. 1985.
37. M. Bramley, E. Burt, K. Corbitt, and B. Kendzierski *et al.* *User's Guide to RC-SToolLBox Version 2.0*. Technical Report 168, Massachusetts Institute of Technology Lincoln Laboratory, September 1991. Lincoln Manual 168.
38. J.W. Brewer. "Kronecker products and matrix calculus in systems theory," *IEEE Trans. on Circuits and Systems*, 25:772-781 (September 1978).
39. Ted J. Broida. "Performance prediction for multi-sensor tracking systems: kinematic accuracy and data association performance." *Sensor Fusion II: Human and Machine Strategies, Proc. SPIE 1198*. 256-271. 1989.
40. W.M. Brown and C.W. Swonger. "A Prospectus for Automatic Target Recognition," *IEEE Trans. Aerospace and Electronic Systems*, 25(3):401-409 (May 1989).
41. B. Burg and B. Zavidovique. "Pattern Recognitions and Image Compression by Means of a Time Warping Algorithm." *Proceedings of the 8th Int'l Cong. on Pattern Recognition*. 762-765. 1986. (CH2342-4/86 1986 IEEE).
42. Rudolph G. Buser, editor. *Proceedings of the First National Symposium on Sensors and Sensor Fusion (1988)*, Chicago, IL: GACIAC, IIT Research Institute, April 1988.
43. Rudolph G. Buser, editor. *Proceedings of the Second National Symposium on Sensors and Sensor Fusion (1989)*, Chicago, IL: GACIAC, IIT Research Institute, March 1989.
44. Rudolph G. Buser, editor. *Proceedings of the Third National Symposium on Sensors and Sensor Fusion (1990)*, Ann Arbor, MI: ERIM, April 1990.
45. C.I. Byrnes *et al.*, editor. *Analysis and Control of Nonlinear Systems*. North Holland: Elsevier Science Publisher B.V., 1988. (see article by Phelps and Krener entitled "Computation of Observer Normal Form Using Macsyma").
46. Neal A. Carlson, R.D. Percy, and S.H. Musick. *Program Design Description for a Multimode Simulation for Optimal Filter Evaluation (MSOFE)*. Technical Report AFWAL-TR-88-1137, Avionics Laboratory, Air Force Wright Aeronautics Laboratory, February 1989.
47. David Casasent. "Advanced Optical Processors for Multiple Degree-of-Freedom Object Recognition," *IEEE Trans. Aerospace and Electronic Systems*, 24(5):608-617 (January 1988).

48. Frank R. Castella. "Tracking Accuracies with Position and Rate Measurements," *IEEE Trans. AES*, 17(3):433-437 (May 1981).
49. N.F. Chamberlain. "Syntactic Classification of Radar Targets Using Polarimetric Signatures." *Proceedings of the 1990 IEEE International Conference on Systems Engineering*. 490-494. 1990.
50. Chaw-Bing Chang and John A. Tabaczynski. "Application of State Estimation to Target Tracking," *IEEE Trans. Automatic Control*, 29(2):98-109 (February 1984).
51. C.B. Chang *et al.* "Application of the Fixed-Interval Smoother to Maneuvering Trajectory Estimation," *IEEE Trans. Automatic Control*, 22(5):876-879 (October 1977).
52. Chung-Ching Chen and Harry C. Andrews. "Target-Motion-Induced Radar Imaging," *IEEE Trans. Aerospace and Electronic Systems*, 16(1):2-14 (January 1980).
53. B.M. Chiesa, S.A. Roth, and D.W. Hill. *High- and Low-Frequency Bistatic RCS Prediction - User's Manual*. Technical Report RADC-TR-88-63, Vol. III (of four), USAF Materiel Command Rome Air Development Center / Syracuse Research Corp., May 1988.
54. K.C. Chou, A.S. Willsky, A. Benveniste, and M. Basseville. "Recursive and Iterative Estimation Algorithms for Multi-Resolution Stochastic Processes." *Proc. 28th IEEE Conf. on Dec. and Cont.*. December 1989.
55. Marvin N. Cohen. "A survey of radar-based target recognition techniques." *Data Structures and Target Classification, Proc. SPIE 1470*, edited by Vibeke Libby. 233-242. 1991.
56. Marvin N. Cohen. "Variability of ultra-high range resolution radar profiles and some implications for target recognition." *Signal Processing, Sensor Fusion, and Target Recognition, Proc. SPIE 1699*, edited by Vibeke Libby and Ivan Kadar. 256-266. 1992.
57. Charles E. Cook and Marvin Bernfeld. *Radar Signals*. New York: Academic Press, 1967.
58. William M. Crowe and James C. Kirsch. "Optical Correlator Guidance Technology Demonstration." *Digital and Optical Shape Representation and Pattern Recognition, Proc. SPIE 938*. 29-35. 1988.
59. S.J. Cusumano and M. De Ponte Jr. *An Extended Kalman Filter Fire Control System Against Air-to-Air Missiles*. MS thesis, U.S. Air Force Institute of Technology, 1977.
60. Frederick E. Daum. "Bounds on Track Purity for Multiple Target Tracking." *Proceedings of the IEEE Conference on Decision and Control*. 1423-1424. 1989.
61. Frederick E. Daum. "Bounds on Performance for Multiple Target Tracking," *IEEE Trans. on Automatic Control*, 35(4):442-446 (April 1990).
62. Frederick E. Daum. "Fundamental Limits in Multisensor Data Fusion." *Proceedings of the 1990 IEEE International Conference on Systems Engineering*. 316-319. 1990.



63. Dave Dayton, Kelly Asher, and Timothy Schneeberger. "Tracker Performance Improvement Using Multiple-Sensor Data." *Sensor Fusion II, Proc. SPIE 1100*. 117-125. 1989.
64. Paul H. Deitz, Michael J. Muuss, and Edwin O. Davisson. "Issues in automatic object recognition: Linking geometry and material data to predictive signature codes." *Automatic Object Recognition IS-7*. 40-56. SPIE Inst. for Advanced Technologies, 1991.
65. Paul H. Deitz et al. *The Ballistic Research Laboratory CAD Package Release 4.0*. Technical Report 168, The U.S. Army Ballistic Research Laboratory, December 1991. (Includes Vols. I-V).
66. Kerim Demirbaş. "Maneuvering -target tracking with the Viterbi algorithm in the presence of interference," *IEE Proceedings*, 136, Pt. F(6):262-268 (December 1989).
67. Eric V. Denardo. *Dynamic Programming: Models and Applications*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1982.
68. Mark R. DeWitt. *High Range Resolution Radar Target Identification Using The Prony Model and Hidden Markov Models*. MS thesis, U.S. Air Force Inst. of Tech., 1992.
69. Hiroshi Dohi and Mitsuru Ishizuka. "A 3-D Vision System Incorporating Solid Modeler and Geometric Reasoning." *Proceedings of the 10th Int'l Cong. on Pattern Recognition I*, edited by Herbert Freeman. 185-187. 1990. (IEEE).
70. Edward R. Dougherty and Charles R. Giardina. *Mathematical Methods for Artificial Intelligence and Autonomous Systems*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1988.
71. Stewart E. Dreyfus and Averill M. Law. *The Art and Theory of Dynamic Programming*. New York: Academic Press, Inc., 1977.
72. Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. New York: John Wiley and Sons, 1973.
73. S.A. Dudani. "Aircraft identification by moment invariants," *IEEE Trans. on Computers*, 26(1):39-46 (January 1977).
74. K.-P. Dunn. *Lower Bounds on Acceleration Estimation Accuracy*. Technical Report 790, Massachusetts Institute of Technology Lincoln Laboratory, October 1987. DTIC No. ADA 188-383.
75. Paul J. Eagle et al. *Target Trajectory Tracking and Prediction Using Regression Dynamics*. Technical Report Contract No. DAAA21-91-C-0085, University of Detroit Mercy, June 1992. (Tech. monitor: Dr. Frank Kuhl, U.S. Army Picatinny Arsenal).
76. Jerry L. Eaves and Edward K. Reedy. *Principles of Modern Radar*. New York: Van Nostrand Reinhold Co., 1987. (see especially Ch. 21 by N.F. Ezquerra on Target Recognition).
77. Manfred Ehrhardt. "Estimation of Aspect Angles by Kalman Filtering With Respect To Electro-Optical Target Identification." *Proc. IEEE NAECON*. 407-414. 1987.

78. Edwin H. Epperson. "Synthetic Generation of Ultra-High Range Resolution (UHRR) Aircraft Signatures." *1990 Joint Service Combat Identification System Conference Technical Proceedings (U)*, edited by Dean Youngberg (Chairman). 465-497. 1990. (article is SECRET/NOFORN: information bearing this citation taken from unclassified pages).
79. Bernard Etkin. *Dynamics of Atmospheric Flight*. New York: John Wiley and Sons, Inc., 1972.
80. N.F. Ezquerra. "Conversation with the Author Regarding Radar Target Recognition." (on 5 Nov., stating to best of his knowledge that 1978 ideas by F. Le Chevalier have not been further developed), 1991.
81. Nabil H. Farhat. "Microwave Diversity Imaging and Automated Target Identification Based on Models of Neural Networks," *Proc. IEEE*, 77(5):670-680 (May 1989).
82. Manuel F. Fernandez. "Data Association Techniques for Multisensor Fusion." *IEEE Workshop on Languages for Automation*. 277-281. 1985.
83. Kenneth H. Fielding *et al.* "Spatio-temporal Pattern Recognition Using Hidden Markov Models." *Proc. SPIE2032*, edited by TBD. TBD. 1993. (to be published).
84. Robert J. Fitzgerald. "Simple Tracking Filters: Steady-State Filtering and Smoothing Performance," *IEEE Trans. Aerospace and Electronic Systems*, 16(6):860-864 (November 1980). (see correction in IEEE TAES, Mar. 1981, pg. 305).
85. Robert J. Fitzgerald. "Simple Tracking Filters: Closed-Form Solutions," *IEEE Trans. Aerospace and Electronic Systems*, 17(6):781-785 (November 1981).
86. Roger Ford. "State of the Art... Steering Tracked Vehicles," *Armed Forces*, 4(5):195-196 (May 1985).
87. R. Frezza and J. Krener *et al.* "Application of an Efficient Nonlinear Filter," *Analysis and Control of Nonlinear Systems*, 223-237 (1988).
88. King-Sun Fu. *Syntactic Methods in Pattern Recognition*. New York: Academic Press, 1974.
89. King-Sun Fu. "Recent Developments in Pattern Recognition," *IEEE Trans. Comput.*, 29(10):845-854 (October 1980).
90. King-Sun Fu. *Syntactic Pattern Recognition and Applications*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1982.
91. Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. New York: Academic Press, 1972.
92. Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition* (Second Edition). Boston, MA: Academic Press, 1990.
93. Jack D. Gaskill. *Linear Systems, Fourier Transforms, and Optics*. New York: John Wiley and Sons, 1978.

94. Author Unknown General Dynamics. *Automatic Radar Target Identification (ARTI) Phase IIIA: Algorithm Q User's Manual*. Technical Report Contract F33615-89-C-1099, Item 0001, CDRL 018-A1D, General Dynamics, September 1991. (provided to author by R. Mitchell, WL/AARA).
95. Bruce P. Gibbs and David W. Porter. "Development and Evaluation of an Adaptive Algorithm for Predicting Tank Motion." *Proceedings of the 19th IEEE Conference on Decision and Control*. 560-567. December 1980.
96. J.M. Gillberg, B.R. Suresh, and D.J. Horowitz. "MMW/IR/LADAR Sensor Fusion for Automatic Targeting." *Proc. 1st Nat'l Symposium on Sensor Fusion II*, edited by Dr. Rudolf J. Buser (Chairman). 531-545. April 1988. (GACIAC PR 88-01).
97. John F. Gilmore. "Knowledge-based target recognition system evolution," *Optical Engineering*, 30(5):557-569 (May 1991).
98. Gonzalez and Wintz. *Digital Image Processing*. Reading, MA: Addison-Wesley, Inc., 1987.
99. I.R. Goodman. "Applications of a Conditional Event Algebra to Data Fusion: Part II." *1989 Tri-Service Data Fusion Symposium Technical Proceedings I*, edited by Otto Kessler (Coordinator). 181-193. 1989. (Vol. I is unclassified).
100. John Gorman. "Conversations with the Author Regarding His Previous Efforts." (stated that he had not applied dynamic time warping techniques to targets with changing aspect angle), September 1991.
101. John W. Gorman, O. Robert Mitchell, and Frank P. Kuhl. "Partial Shape Recognition Using Dynamic Programming," *IEEE Trans. Patt. Analy. and Mach. Intel.*, 10(2):257-266 (March 1988).
102. Paul G. Gottschalk *et al.* "Recognition, Tracking, and Pose Estimation of Arbitrarily Shaped 3-d Objects in Cluttered Intensity and Range Imagery." *Sensor Fusion III: 3-D Perception and Recognition, Proc. SPIE1383*, edited by Paul S. Schenker. 84-96. 1990.
103. Don A. Gregory and Larry Z. Kennedy. "Computer generated matched filters with spatial frequency tailoring." *Optical and Digital Pattern Recognition, Proc. SPIE754*. 314-318. 1988.
104. Richard S. Grote and Jerome H. Hutcheson. *Aircraft Survivability Conceptual Design Criteria, Vol. II*. Technical Report AFWAL-TR-82-3062, La Jolla, CA and WPAFB, OH: Science Applications International Corp. and USAF Flight Dynamics Laboratory, 1982.
105. Jay K. Hackett, Matt J. Lavoie, and Mubarak Shah. "Three-dimensional Object Recognition Using Multiple Sensors." *Sensor Fusion III: 3-D Perception and Recognition, Proc. SPIE1383*, edited by Paul S. Schenker. 611-622. 1990.
106. Jay K. Hackett and Mubarak Shah. "Multi-Sensor Fusion: A Perspective." *Proceedings of the 1990 IEEE Conference on Robotics and Automation*. 1324-1330. 1990.
107. David L. Hall. *Techniques for Multisensor Data Fusion*. Norwood, MA: Artech House, 1992.

108. Yu-Chi Ho and Ashok K. Agrawala. "On Pattern Classification Algorithms: Introduction and Survey," *Proceedings of the IEEE*, 56(12):2101-2114 (December 1968).
109. Moshe Horev. *Picture Correlation Model for Automatic Target Recognition*. MS thesis, U.S. Air Force Inst. of Tech., 1980.
110. Berthold Klaus Paul Horn. *Robot Vision*. Cambridge, MA: The MIT Press, 1986.
111. M.K. Hu. "Visual pattern recognition by moment invariants," *IEEE Trans. Inform. Theory*, 8:179-187 (February 1962).
112. Klaus Huenecke. *Modern Combat Aircraft Design*. Shrewsbury, England: Airlife Publishing Ltd., 1987.
113. Charles E. Hutchinson. "The Kalman Filter Applied to Aerospace and Electronic Systems," *IEEE Trans. AES*, 20(4):500-504 (July 84).
114. IMSL. *User's Manual, IMSL Math/Library*. Houston, TX, 1989.
115. Fumitada Itakura. "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. ASSP*, 23(1):67-72 (February 1975).
116. Siu-Leong Iu and Kwangyoen Wahn. "Estimation of 3-D Motion and Structure Based on a Temporally-Oriented Approach With the Method of Regression." *IEEE Workshop on Visual Motion*. 273-281. March 1989.
117. Siu-Leong Iu and Kwangyoen Wahn. "Estimation of General Rigid Body Motion from a Long Sequence of Images." *Proceedings of the 10th Int'l Cong. on Pattern Recognition I*, edited by Herbert Freeman. 217-219. 1990. (IEEE).
118. T. Kailath. "An Innovations Approach to Least Squares Estimation - Part I: Linear Filtering in Additive White Noise," *IEEE Trans. Automatic Control*, 13(6):646-655 (December 1968).
119. R.E. Kalman. "A new approach to linear filtering and prediction problems," *J. Basic Engng., Trans. ASME, Ser. D*, 82:35-45 (March 1960).
120. Jerry D. Kendrick. *Estimation of Aircraft Target Motion Using Pattern Recognition Orientation Measurements*. PhD dissertation, U.S. Air Force Inst. of Tech., 1978.
121. Jerry D. Kendrick, P.S. Maybeck, and J.G. Reid. "Estimation of Aircraft Target Motion Using Orientation Measurements," *IEEE Trans. AES*, 17(2):254-259 (March 1981).
122. Stephen C. Kenyon. "Fusion of Target Features and Target Dynamics to Enhance Multitarget Tracking." *Proc. 1st Nat'l Symposium on Sensor Fusion II*, edited by Dr. Rudolf J. Buser (Chairman). 423-431. April 1988. (GACIAC PR 88-01).
123. Steve Kenyon. "Conversations with the Author Regarding His Previous Efforts." (stated that his 1988 article did not imply "motion warping"), September 1991.
124. R.J. Kenyon (Chairman), editor. *1989 Tri-Service Combat Identification Systems Conference Technical Proceedings (U)*, Warminster, PA: Naval Air Development Center, June 1989. (Classified SECRET/NOFORN).
125. Thomas Kerr. "Status of CR-Like Lower Bounds for Nonlinear Filtering," *IEEE Trans. AES*, 25(5):590-601 (September 1989).

126. Otto Kessler (Coordinator), editor. *1989 Tri-Service Data Fusion Symposium Technical Proceedings*, Warminster, PA: Naval Air Development Center, June 1989. (Vol. I unclassified, Vol. II classified SECRET).
127. William F. Kobel and Timothy Martin. *Distortion-Invariant Pattern Recognition in Non-Random Noise*. MS thesis, U.S. Air Force Inst. of Tech., 1986.
128. Eric T. Kouba. *Recurrent Neural Networks for Radar Target Recognition*. MS thesis, U.S. Air Force Inst. of Tech., 1992.
129. J. David R. Kramer Jr. and Wallace S. Reid. "Track-Before-Detect Processing for an Airborne Type Radar." *Proceedings of the IEEE 1990 International Radar Conference*. 422-427. 1990.
130. A.J. Krener. "The Asymptotic Approximation of Nonlinear Filters by Linear Filters," *Theory and Applications of Nonlinear Control Systems*, 359-378 (1986).
131. A.J. Krener and A. Isadori. "Linearization by output injection and nonlinear observers," *Systems and Control Letters*, 3(1):47-52 (1983).
132. A.J. Krener and W. Respondek. "Nonlinear Observers with Linearizable Error Dynamics," *SIAM J. Control and Optimization*, 23(2):197-215 (March 1985).
133. R.E. Larson and J. Peschon. "A Dynamic Programming Approach to Trajectory Estimation," *IEEE Trans. Automatic Control*, 11(3):537-540 (July 1966).
134. François Le Chevalier. "Target and Background Signatures and Radar Signal Processing," *La Recherche Aerospatiale*, (1) (January 1986). (published in English and French editions).
135. François Le Chevalier and Gerard Bobillot, "Real-time syntactic signal from noise separation." French patent, 1979. (French patent application number 7727362, 9 Sept. 1977; final French patent number 2402971 A 790511 7924; UK patent application number 2 010 626 A, 11 Sept. 1978).
136. François Le Chevalier, Gerard Bobillot, and Cecile Fugier-Garrel. "Radar Target and Aspect Angle Identification." *Proceedings of the IEEE 1978 International Conference on Pattern Recognition*. 398-400. 1978.
137. François Le Chevalier *et al.* "Etude des Possibilites de Classification des Cibles par Radar Aeroporte." *AGARD CP-290: Image and Sensor Data Processing for Target Acquisition and Recognition (Classified Supplement)*, edited by R. Voles. 1980.
138. François Le Chevalier *et al.* "An Approach to Recognition Criteria of Radar or Sonar Targets." *Proceedings of the 1984 International Conference on Radar*. 269-274. 1984. (not an IEEE-sponsored conference).
139. Sukhan Lee and Youngchul Kay. "A Kalman Filter Approach for Accurate 3-D Motion Estimation from a Sequence of Stereo Images." *Proceedings of the 10th Int'l Cong. on Pattern Recognition*, edited by Herbert Freeman. 104-108. 1990. (IEEE).
140. C.T. Leondes, editor. *Control and Dynamic Systems*, 261-302. New York: Academic Press, Inc., 1988, Part 1. (Vol. 28: see article by Sworder and Hutchins).

141. C.T. Leondes, editor. *Control and Dynamic Systems*, 273-305. New York: Academic Press, Inc., 1990. (Vol. 38: see article by Sworder and Singer).
142. Michel Lévêque and François Le Chevalier *et al.*, "System for Automatically Guiding a Missile and Missile Provided With Such a System." U.S. Patent, 1988. (U.S. patent number 4,735,379).
143. Edmund W. Libby. *Development of an Optimal Estimator for Terminal Guidance of an Air-to-Ground Radar-Guided Missile*. MS thesis, U.S. Air Force Inst. of Tech., 1983.
144. Peter W. Likins. *Elements of Engineering Mechanics*. New York, NY: McGraw-Hill Book Co., 1973.
145. R.J. Linn, D.L. Hall, and J. Llinas. "A Survey of Multi-Sensor Data Fusion Systems." *Data Structures and Target Classification, Proc. SPIE 1470*. 13-29. 1991.
146. Ren C. Luo and Michael G. Kay. "Multisensor Integration and Fusion in Intelligent Systems," *IEEE Trans. on Systems, Man, and Cybernetics*, 19(5):901-931 (Sept./Oct. 1989).
147. D.T. Magill. "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. Automatic Control*, 10(5):434-439 (1965).
148. A.K. Mahalanabis, S. Prasad, and A. Garg. "A Smoothing Algorithm for Improved Tracking in Clutter and Multitarget Environment." *IEEE Automatic Control Conference*. 1107-1112. 1986. paper TA8 - 10:45.
149. A.K. Mahalanabis and Bin Zhou. "A Joint Probabilistic Data Association Smoothing Algorithm for Multitarget Tracking." *IEEE Automatic Control Conference*. 430-435. 1988. paper WP1 - 4:30.
150. S.G. Mallat. *A theory for multiresolution signal decomposition: the wavelet representation*. Technical Report MS-CIS-87-22, Dept. of Computer and Info. Science - Univ. of Penn., May 1987.
151. Mathworks. *Pro-Matlab for Sun Workstations*. South Natick, MA, 1990.
152. Peter S. Maybeck. *Combined State and Parameter Estimation for On-Line Applications*. PhD dissertation, Massachusetts Institute of Technology, 1972.
153. Peter S. Maybeck. *Stochastic Models, Estimation, and Control: Volume 1*. New York: Academic Press, 1979.
154. Peter S. Maybeck. *Stochastic Models, Estimation, and Control: Volume 2*. New York: Academic Press, 1982.
155. Peter S. Maybeck. *Stochastic Models, Estimation, and Control: Volume 3*. New York: Academic Press, 1982.
156. Peter S. Maybeck, W.H. Worsley, and P.M. Flynn. "Investigation of Constant Turn-Rate Dynamics Models in Filters for Airborne Vehicle Tracking." *Proc. IEEE NAECON*. 892-903. 1982.

157. P.S. Maybeck, S.J. Cusumano, M. De Ponte Jr., and J.E. Negro. "Enhanced Fire Control System Filtering Via Refined Air-to-Air Missile Acceleration Modelling," *Proc. 1978 IEEE Conf. on Decision and Control*, 80-87 (January 1979).
158. Dan Jr. McLachlan. "The Role of Optics in Applying Correlation Functions to Pattern Recognition," *Journal of the Optical Society of America*, 52(4):454-459 (April 1962).
159. J.S. Meditch. *Stochastic Linear Optimal Estimation and Control*. New York: McGraw-Hill, 1969.
160. J.S. Meditch. "A Survey of Data Smoothing for Linear and Nonlinear Dynamic Systems," *Automatica*, 9:151-162 (April 1973).
161. Laurent Miclet. *Structural Models in Pattern Recognition*. New York: Springer-Verlag, 1986. (translated by J. Howlett from 1984 French edition).
162. Harry Mieras. "Conversations with the Author Regarding Raytheon Efforts on Dynamic Programming for Integration of High Range Resolution Radar Data." (discussed article in 1990 CISC Proceedings), December 1991.
163. Harry Mieras. "Letter to LTC Libby regarding review of draft journal article on the Libby/Maybeck research." (response to letter requesting review of subject article), March 1993.
164. Harry Mieras and Timothy Cichocki *et al.* "Airframe Identification Results Using High Range Resolution." *1990 Joint Service Combat Identification System Conference Technical Proceedings (U)*, edited by Dean Youngberg (Chairman). 1107-1115. 1990. (SECRET/NOFORN).
165. H. Mieras *et al.* *Non-Cooperative Target Recognition - High Range Resolution (NCTR-HRR) Algorithm Evaluation (U)*. Technical Report BR-19437, Raytheon, February 1991. (SECRET report: obtainable from DTIC).
166. Rick Mitchell *et al.* "Conversations with the Author Regarding High Range Resolution Radar Target Recognition." (input from Wright Laboratory Target Recognition Office), 1991.
167. Stanton H. Musick and Neal A. Carlson. *User's Manual for a Multimode Simulation for Optimal Filter Evaluation (MSOFE)*. Technical Report AFWAL-TR-88-1138, Avionics Laboratory, Air Force Wright Aeronautics Laboratory, April 1990.
168. Cory Myers, Lawrence R. Rabiner, and Aaron E. Rosenberg. "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition," *IEEE Trans. ASSP*, 28(6):623-635 (December 1980).
169. Hatem N. Nasr (Ed.). *Automatic Object Recognition, IS-7*. Bellingham, WA: SPIE Inst. for Advanced Technologies, 1991.
170. Arch W. Naylor and George R. Sell. *Linear Operator Theory in Engineering and Science*. New York: Springer Verlag, 1982.
171. S.R. Neal. "Discussion on "Parametric Relations for the  $\alpha - \beta - \nu$  Filter Predictor"," *IRE Transactions on Automatic Control*, 315-317 (June 1967).

172. V.G. Nebabin and V.V. Sergeyev. *Radar Recognition: Methods and Techniques*. Moscow, Russia: Radio I Svyaz Publishing House, 1984. (Original is in Russian: English translation available from DTIC, No. AD-B094 343).
173. Lawrence C. Ng and Yaakov Bar-Shalom. "Multisensor Multitarget Time Delay Vector Estimation," *IEEE Trans. ASSP*, 34(4):669-678 (August 1986).
174. Yoh-Han Pao. *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley Pub. Co., 1989.
175. Robert M. Pap *et al.* "Neural Network Implementation of Multisensor Fusion for Target Identification Problems." *Proc. 3d Nat'l Symposium on Sensor Fusion II*. 135-157. 1990. (unclassified article in classified volume).
176. Thomas W. Parsons. *Voice and Speech Processing*. New York: McGraw-Hill, Inc., 1987.
177. Judea Pearl. "Reverend Bayes on Inference Engines: a Distributed Heirarchical Approach." (*IEEE*) *Proc. Second National Conf. on Artificial Intelligence*. 133-136. 1982.
178. William K. Pratt. *Digital Image Processing* (Second Edition). New York: John Wiley and Sons, 1991.
179. Keith Price. "Multi-Frame Feature-Based Motion Analysis." *Proceedings of the 10th Int'l Cong. on Pattern Recognition*, edited by Herbert Freeman. 114-117. 1990. (IEEE).
180. L.R. Rabiner. "Telephone Conversation with the Author Regarding Application of Dynamic Time Warping to Multisensor Fusion and Radar Tracking." (stating to the best of his knowledge that time warping techniques had not been applied to radar signal processing or multisensor fusion), October 1991.
181. L.R. Rabiner. "Telephone Conversation with the Author." (discussed sequence length mismatches and local normalization in dynamic programming sequence comparison), December 1992.
182. L.R. Rabiner, A.E. Rosenberg, and S.E. Levinson. "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," *IEEE Trans. ASSP*, 26(6):575-582 (December 1978).
183. L.R. Rabiner and Carolyn E. Schmidt. "Application of Dynamic Time Warping to Connected Digit Recognition," *IEEE Trans. ASSP*, 28(4):377-388 (August 1980).
184. C.R. Rao. *Linear Statistical Inference and Its Applications*. New York: John Wiley and Sons, Inc., 1973.
185. S.K. Rogers, Dennis Ruck, and Matthew Kabrisky. *Artificial Neural Nets*. WPAFB, OH: U.S. Air Force Inst. of Technology, 1991.
186. Michael C. Roggemann. *Multiple Sensor Fusion for Detecting Targets in FLIR and Range Images*. PhD dissertation, U.S. Air Force Institute of Technology, 1989.
187. J. Roskam. *Airplane Flight Dynamics and Automatic Control, Part 1*. Lawrence, KS: Roskam Aviation and Engineering Corp. and the University of Kansas, 1979.



188. William E. Ross and Stuart A. Mills. "Miniature Ruggedized Optical Correlator." *1990 Joint Service Combat Identification System Conference Technical Proceedings (U)*, edited by Dean Youngberg (Chairman). 843-863. 1990. (article is unclassified).
189. Peter L. Rothman and Richard V. Denton. "Fusion or confusion: knowledge or nonsense." *Data Structures and Target Classification, Proc. SPIE1470*. 2-12. 1991.
190. Dennis Ruck. *Characterization of Multilayer Perceptrons and their Application to Multisensor Automatic Target Detection*. PhD dissertation, U.S. Air Force Institute of Technology, 1990.
191. Firooz A. Sadjadi and Michael Bazakos. "A perspective on automatic target recognition evaluation technology," *Optical Engineering*, 30(2):141-146 (February 1991).
192. H. Sakoe and S. Chiba. "A dynamic programming approach to continuous speech recognition." *Proc. Int. Congr. Acoust.*. 1971. (paper 20C-13).
193. H. Sakoe and S. Chiba. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Trans. ASSP*, 26(1):43-49 (February 1978).
194. O.S. Sands and F.D. Garber. "Pattern Representations and Syntactic Classification of Radar Measurements of Commercial Aircraft," *IEEE Trans. PAMI*, 12(2):204-211 (February 1990).
195. David Sankoff and Joseph B. Kruskal. *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*. Reading, Massachusetts: Addison-Wesley Publishing Co., Inc., 1983.
196. Louis L. Scharf and Howard Elliott. "Aspects of Dynamic Programming in Signal and Image Processing," *IEEE Trans. on Automatic Control*, 26(5):1018-1029 (October 1981).
197. Richard L. Scheaffer. *Introduction to Probability and Its Applications*. Boston, MA: PWS-KENT Publishing Co., 1990.
198. F.C. Schweppe. *Uncertain Dynamic Systems*. Englewood Cliffs, NJ: Prentice Hall, 1973.
199. Michael Seibert and Allen M. Waxman. "Combining Evidence From Multiple Views of 3-D Objects." *Sensor Fusion IV, Proc. SPIE1611*, edited by Paul S. Schenker. 178-189. 1991.
200. Hemchandra M. Shertukde and Yaakov Bar-Shalom. "Tracking of Crossing Targets with Imaging Sensors," *IEEE Trans. AES*, 27(4):582-592 (July 1991).
201. H.R. Simpson. "Performance Measures and Optimization Condition for a Third-Order Sampled-Data Tracker," *IRE Transactions on Automatic Control*, 182-183 (April 1963).
202. R.A. Singer. "Estimating Optical Tracking Filter Performance for Manned Maneuvering Targets," *IEEE Trans. AES*, 6(4):473-483 (July 1970).
203. Merrill I. Skolnik. *Introduction to Radar Systems* (Second Edition). New York: McGraw-Hill Book Company, 1980.
204. Merrill I. Skolnik. *Radar Handbook*. New York: McGraw-Hill Book Company, 1990.

205. D.L. Snyder. *Random Point Processes*. New York: John Wiley and Sons, Inc., 1975.
206. Tarek M. Sobh and Kwangyoen Wohn. "Recovery of 3-D Motion and Structure by Temporal Fusion." *Sensor Fusion II: Human and Machine Strategies, Proc. SPIE1198*. 147-155. 1989.
207. Clayton Stewart, Victor Larson, and J. Doss Halsey. "Comparison of classification approaches for high-range resolution radar." *Automatic Object Recognition II, Proc. SPIE1700*. 146-155. 1992.
208. D.D. Sworder and R. G. Hutchins. "Image-Enhanced Tracking," *IEEE Trans. AES*, 25(5):701-710 (September 1989).
209. D.D. Sworder and R. G. Hutchins. "Maneuver Estimation Using Measurements of Orientation," *IEEE Trans. AES*, 26(4):625-637 (July 1990).
210. J.H. Taylor. "The Cramér-Rao Estimation Error Lower Bound Computation for Deterministic Nonlinear Systems," *IEEE Trans. Auto. Control* (April 1979).
211. Charles W. Therrien. "A Sequential Approach to Target Discrimination," *IEEE Trans. AES*, 14(3):433-440 (May 1978).
212. J.T. Tou and R.C. Gonzalez. *Pattern Recognition Principles*. Reading, MA: Addison-Wesley, 1974.
213. Dan A. Turner and Walter N. Barnes. "Ultra High Range Resolution (UHRR) for Air-to-Air Target ID (U)." *1989 Tri-Service Combat Identification Systems Conference Technical Proceedings (U)*. 121-128. June 1989. (text is unclassified - graphs are secret).
214. V.M. Velichko and N.G. Zagoruyko. "Automatic Recognition of 200 Words," *Int. Jour. Man-Machine Studies*, 2:223 (June 1970).
215. B.L. Walcott, M.J. Corless, and S.H. Zak. "Comparative study of non-linear state observation techniques," *Int. Jour. Control*, 45(6):2109-2132 (1987).
216. A. Wald. *Sequential Analysis*. New York: John Wiley and Sons, Inc., 1947.
217. T.P. Wallace and P.A. Wintz. "An efficient three-dimensional aircraft recognition algorithm using Fourier descriptors," *Comput. Graphics Image Processing*, 13:99-126 (1980).
218. Edward Waltz and James Llinas. *Multisensor Data Fusion*. Norwood, MA: Artech House, 1990.
219. Philip D. Wasserman. *Neural Computing: Theory and Practice*. New York: Van Nostrand Reinhold, 1989.
220. H. Wechsler, editor. *Neural Networks for Human and Machine Perception*. New York: Academic Press, Inc., 1991. (see chapter by Seibert and Waxman).
221. Donald R. Wehner. *High Resolution Radar*. Norwood, MA: Artech House, 1987.
222. Anthony J. Weiss and Benjamin Friedlander. "Efficient Dynamic Programming in the Presence of Nuisance Parameters," *IEEE Trans. AES*, 25(2):277-280 (March 1989).

223. G.M. White and R.B. Neely. "Speech recognition experiments with linear prediction, bandpass filtering, and dynamic programming," *IEEE Trans. ASSP*, 24(2):183-188 (April 1976).
224. F. White (Chairman), "1989 Data Fusion Survey." Unclassified annex to classified proceedings for the 1990 Data Fusion Symposium at Johns Hopkins Univ., May 1990.
225. Donald Willis and Brian Singstock. "Conversations with the author regarding current contractor activities in target recognition." (included review of contractors' documents), August 1991.
226. Donald J. Willis. *Feature Extraction for Pose Estimation: A Comparison Between Synthetic and Real Imagery*. MS thesis, U.S. Air Force Inst. of Tech., 1991.
227. John P. Wojnar, Dale E. Bardin, and Ben Hansburg. "A Dual Level Data Fusion Methodology For Multi-Sensor Autonomous Target Recognition." *Proc. 1st Nat'l Symposium on Sensor Fusion II*, edited by Dr. Rudolf J. Buser (Chairman). 319-340. April 1988. (GACIAC PR 88-01).
228. Derek Wood, editor. *Jane's World Aircraft Recognition Handbook* (Fourth Edition). Surrey, UK: Jane's Information Group Ltd., 1989.
229. William H. Worsley. *Comparison of Three Extended Kalman Filters for Air-to-Air Tracking*. MS thesis, U.S. Air Force Inst. of Tech., 1979.
230. Hiromitsu Yamada and Kazuhiko Yamamoto. "Recognition of Echocardiograms by Dynamic Programming Matching Method." *Proceedings of the 9th Int'l Cong. on Pattern Recognition*. 685-688. 1988. (IEEE).
231. Dean Youngberg (Chairman), editor. *1990 Joint Service Combat Identification System Conference Technical Proceedings (U)*, Warminster, PA: Naval Air Development Center, December 1990. (Classified SECRET/NOFORN).
232. Francis T.S. Yu *et al.* "Rotation invariant pattern recognition with a programmable joint transform correlator," *Applied Optics*, 28(22):4725-4727 (November 1989).



**SUPPLEMENTARY**

**INFORMATION**

LTC Edmund W. Libby  
5435 Rossell Loop  
Ft. Belvoir, VA 22060

UMI Dissertation Services  
Attn.: Ms. ~~Jerrie~~ Gray  
1490 Eisenhower Ave.  
P.O. Box 975  
Ann Arbor, MI 48104

*Mr. Banks*

Dear Ms. ~~Gray~~: *Mr. Banks*

My dissertation, entitled "Application of Sequence Comparison Methods to Multisensor Data Fusion and Target Recognition" was sent to your firm in July or August of 1993 from the U.S. Air Force Institute of Technology at Wright-Patterson Air Force Base (Dayton), Ohio.

The purpose of this letter is to submit a change to page 3-27 of the dissertation. The new page is attached at Encl. 1. Also attached are a letter from my committee chairman, Prof. Peter S. Maybeck, authorizing this change, and a money order for \$25.00, which I am told is your charge for a negative correction.

I can be reached at (703) 697-7645/0638/0854 for further information. Prof. Maybeck can be reached at (513) 255-3708, Ext. 4639.

Thank You,

*E. W. Libby*

Edmund W. Libby

ERRATA ADA 267623

Peschon, and then we will consider modifications that might be desired to provide more rigorous answers, if the basic assumptions of this development do not apply.

*3.6.3 Relating  $I^*(\mathbf{x}_k^a, k)$  to Object Class Probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ .* Recall that the "practical version" of the Larson and Peschon method as implemented on different object models  $\omega_i$  would find the  $\mathbf{X}_{k,n}^a$ , say  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ , for each  $\omega_i$ , that maximizes  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ , not by maximizing this conditional probability directly, but rather by maximizing  $I^*(\mathbf{x}_k^a, k)$ , where (using the original Larson and Peschon form, but recalling that in our development for object recognition, all probabilities would be conditioned also on  $\mathbf{Z}_m^d$  and  $\omega_i$ ):

$$\text{MAX}_{\mathbf{x}_k^a} I^*(\mathbf{x}_k^a, k) = \text{MAX}_{\mathbf{x}_k^a} \left\{ \text{MAX}_{\mathbf{x}_{k-1}^a} \left[ p(\mathbf{z}_k^f | \mathbf{x}_k^a) p(\mathbf{x}_k^a | \mathbf{x}_{k-1}^a) I^*(\mathbf{x}_{k-1}^a, k-1) \right] \right\} \quad (3.10)$$

or, in our form:

$$\begin{aligned} \text{MAX}_{\mathbf{x}_{k,n}^a} I^*(\mathbf{x}_{k,n}^a, k | \omega_i) = \\ \text{MAX}_{\mathbf{x}_{k,n}^a} \left\{ \text{MAX}_{\mathbf{x}_{k-1,n}^a} \left[ p(\mathbf{z}_k^f | \mathbf{x}_{k,n}^a, \omega_i) p(\mathbf{x}_{k,n}^a | \mathbf{x}_{k-1,n}^a, \mathbf{Z}_m^d, \omega_i) I^*(\mathbf{x}_{k-1,n}^a, k-1 | \omega_i) \right] \right\} \end{aligned} \quad (3.11)$$

Maximizing this quantity rather than the conditional probability is desirable because we avoid having to compute values for all  $\mathbf{X}_{k,n}^a \in \mathcal{X}_{pk}^a$ , which we would have to do to find the denominator term as in Eqn. (3.6). Examining  $I^*(\mathbf{x}_{k,n}^a, k | \omega_i)$  closely, note that the preceding equation is equivalent to:

$$\text{MAX}_{\mathbf{x}_{k,n}^a} I^*(\mathbf{x}_{k,n}^a, k | \omega_i) = \text{MAX}_{\mathbf{X}_{k,n}^a} \left[ p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f | \mathbf{Z}_m^d, \omega_i) \right] \quad (3.12)$$

Thus, the practical version of the Larson and Peschon equations gives the maximum value of  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f | \mathbf{Z}_m^d, \omega_i)$ , and the state history estimate  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$  for a given  $\omega_i$  which gives that maximum joint conditional probability density, or, equivalently, since there is only one  $\mathbf{Z}_k^f$ , that state history which maximizes  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ .

Suppose on the other hand that we had chosen to find  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f, \omega_i | \mathbf{Z}_m^d)$  for all  $\mathbf{X}_{k,n}^a \in \mathcal{X}_{pk}^a$ . This can be had by computing  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f | \mathbf{Z}_m^d, \omega_i)$  for each  $\mathbf{X}_{k,n}^a$  over each  $\omega_i$

**SUPPLEMENTARY**

**INFORMATION**



ERRATA - ADA 267623

6 October 1995

LTC Edmund W. Libby, Ph.D.  
485 Cooke Dr.  
Redstone Arsenal, AL 35808

Defense Technical Information Center  
Attn: RSR (Mr. Banks / Ms. Campbell)  
8725 John J. Kingman Rd.  
Suite 0944  
Ft. Belvoir, VA 22060 - 6218

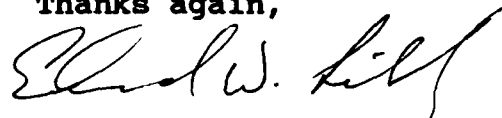
Dear Mr. Banks / Ms. Campbell:

My dissertation, entitled "Application of Sequence Comparison Methods to Multisensor Data Fusion and Target Recognition" (DTIC No. ADA 267623) was sent to DTIC in July or August of 1993 from the U.S. Air Force Institute of Technology at Wright-Patterson AFB (Dayton), Ohio.

Several months later, I submitted a change to page 3-27. Subsequently I realized, however, that this change still contains a minor error -- a comma and a letter "n" are missing in one equation.

Enclosed is a "re-revised" page 3-27. I appreciate your help I making this correction. It's a small item, but I want the document to be correct.

Thanks again,



Edmund W. Libby, Ph.D.  
PM for Theater Targets

ERRATA

AD-A 267623

Peschon, and then we will consider modifications that might be desired to provide more rigorous answers, if the basic assumptions of this development do not apply.

**3.6.3 Relating  $I^*(\mathbf{x}_k^a, k)$  to Object Class Probability  $p(\omega_i | \mathbf{Z}_k^f, \mathbf{Z}_m^d)$ .** Recall that the "practical version" of the Larson and Peschon method as implemented on different object models  $\omega_i$  would find the  $\mathbf{X}_{k,n}^a$ , say  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ , for each  $\omega_i$ , that maximizes  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ , not by maximizing this conditional probability directly, but rather by maximizing  $I^*(\mathbf{x}_k^a, k)$ , where (using the original Larson and Peschon form, but recalling that in our development for object recognition, all probabilities would be conditioned also on  $\mathbf{Z}_m^d$  and  $\omega_i$ ):

$$\text{MAX}_{\mathbf{x}_k^a} I^*(\mathbf{x}_k^a, k) = \text{MAX}_{\mathbf{x}_k^a} \left\{ \text{MAX}_{\mathbf{x}_{k-1}^a} \left[ p(\mathbf{z}_k^f | \mathbf{x}_k^a) p(\mathbf{x}_k^a | \mathbf{x}_{k-1}^a) I^*(\mathbf{x}_{k-1}^a, k-1) \right] \right\} \quad (3.10)$$

or, in our form:

$$\begin{aligned} \text{MAX}_{\mathbf{x}_{k,n}^a} I^*(\mathbf{x}_{k,n}^a, k | \omega_i) = \\ \text{MAX}_{\mathbf{x}_{k,n}^a} \left\{ \text{MAX}_{\mathbf{x}_{k-1,n}^a} \left[ p(\mathbf{z}_k^f | \mathbf{x}_{k,n}^a, \omega_i) p(\mathbf{x}_{k,n}^a | \mathbf{x}_{k-1,n}^a, \mathbf{Z}_m^d, \omega_i) I^*(\mathbf{x}_{k-1,n}^a, k-1 | \omega_i) \right] \right\} \end{aligned} \quad (3.11)$$

Maximizing this quantity rather than the conditional probability is desirable because we avoid having to compute values for all  $\mathbf{X}_{k,n}^a \in \mathcal{X}_{pk}^a$ , which we would have to do to find the denominator term as in Eqn. (3.6). Examining  $I^*(\mathbf{x}_{k,n}^a, k | \omega_i)$  closely, note that the preceding equation is equivalent to:

$$\text{MAX}_{\mathbf{x}_{k,n}^a} I^*(\mathbf{x}_{k,n}^a, k | \omega_i) = \text{MAX}_{\mathbf{X}_{k,n}^a} \left[ p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f | \mathbf{Z}_m^d, \omega_i) \right] \quad (3.12)$$

Thus, the practical version of the Larson and Peschon equations gives the maximum value of  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f | \mathbf{Z}_m^d, \omega_i)$ , and the state history estimate  $\hat{\mathbf{X}}_{k/\omega_i}^{LP}$ , for a given  $\omega_i$  which gives that maximum joint conditional probability density, or, equivalently, since there is only one  $\mathbf{Z}_k^f$ , that state history which maximizes  $p(\mathbf{X}_{k,n}^a | \mathbf{Z}_k^f, \mathbf{Z}_m^d, \omega_i)$ .

Suppose on the other hand that we had chosen to find  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f, \omega_i | \mathbf{Z}_m^d)$  for all  $\mathbf{X}_{k,n}^a \in \mathcal{X}_{pk}^a$ . This can be had by computing  $p(\mathbf{X}_{k,n}^a, \mathbf{Z}_k^f | \mathbf{Z}_m^d, \omega_i)$  for each  $\mathbf{X}_{k,n}^a$  over each  $\omega_i$